

Application Design Issues for Electronic Marketplace

Specifika návrhu aplikaci pro elektronická tržiště

Bc. Václav Dorazil

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Václav Dorazil**
Osobní číslo: **A11478**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Specifika návrhu aplikací pro elektronická tržiště**

Zásady pro vypracování:

1. Seznamte se s doporučeními společnosti Microsoft, které definují moderní aplikace pro platformu Windows.
2. Vytvořte modely popisující funkčnost aplikace.
3. Analyzujte a navrhňte případy užití definující chování aplikace.
4. Vytvořte wireframy definující vzhled aplikace.
5. Proveďte analýzu rizik pro distribuci aplikace.
6. Navrhňte kroky, které povedou k eliminaci nebo snížení vlivu rizik.
7. Vyhodnoťte vhodnost návrhu a použitých metod.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Jakob NIELSEN. Designing Web Usability: The Practice of Simplicity. Berkeley, CA: Peachpit Press, 2000, 419s. ISBN: 1-56205-810-X.**
2. **Drusinsky, Doron. Modeling and verification using UML statecharts : a working guide to reactive system design, runtime monitoring, and execution-based model checking. Oxford : Newnes, 2006, xii, 306 s. ISBN 0750679492.**
3. **Pilone, Dan. UML 2.0 pocket reference. Beijing ; Farnham : O'Reilly, 2006, vii, 128 s. ISBN 978-0-596-10208-1.**
4. **Stellman, Andrew. Applied software project management. Sebastopol, CA : O'Reilly, 2006, xii, 308 s. ISBN 978-0-596-00948-9.**
5. **ARNOWITZ, Jonathan, Michael ARENT a Nevin BERGER. Effective prototyping for software makers. 1st ed. Amsterdam: Elsevier, 2007, xxxviii, 584 s. ISBN 0120885689.**
6. **BROWN, Daniel M. Communicating design: developing web site documentation for design and planning. Berkeley, CA: Peachpit Press, 2007, xiv, 352 s. ISBN 0321392353.**
7. **MILES, Russ a Kim HAMILTON. Learning UML 2.0. 1st ed. Beijing: O'Reilly, 2006, xiii, 269 s. ISBN 0596009828.**

Vedoucí diplomové práce:

Ing. Radek Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

22. února 2013

Termín odevzdání diplomové práce:

22. května 2013

Ve Zlíně dne 22. února 2013

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRACT

The subject of this thesis is to create product specifications for mobile applications designed for distribution in the windows store.

In the theoretical part of this thesis, the project management rules and methodology are which should be used for general software development are described. The Theoretical portion also contains a description of UML as a methodology used for creating specifications, and ASO focuses on the distribution of applications in stores.

The Practical part of this thesis contains exact product specifications for wallpaper applications. Applications in which allow the user to customize Windows 8 operating system by setting the account picture, lock screen, and desktop wallpaper. In addition the practical part describes the risks that can occur with the distribution process and for every risk there are recommended action steps that lead to its elimination.

Keywords: UML, mobile application, windows store, ASO, Windows 8

ABSTRAKT

Předmětem této diplomové práce je vytvořit specifikaci mobilní aplikace určené pro distribuci na Windows Store.

Teoretická část diplomové práce uvádí základy projektového vedení, které jsou platné obecně pro vývoj softwaru. Současně jsou v teoretické části popsány prvky jazyka UML, který se využívá pro návrh aplikací. Na závěr se teoretická část věnuje principům optimalizace pro distribuci mobilních aplikací.

Praktická část obsahuje konkrétní produktovou specifikaci pro aplikaci s obrázky, která umožňuje uživatelům nastavit úvodní obrazovku, profilový obrázek a wallpaper ve Windows 8. Praktická část uvádí také rizika pro distribuci aplikace včetně akčních kroků, které možná rizika minimalizují nebo eliminují.

Klíčová slova: UML, mobilní aplikace, Windows Store, ASO, Windows 8

Acknowledgement:

I would like to thank every one I have been involved with during the course of my studies and every one who has had a hand in helping me create this thesis. I would like to especially thank Mr.Ing.Radek Šilhavý Ph.D. for his attention and professional guidance throughout this process.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

CONTENT

INTRODUCTION	11
I THEORETICAL PART	12
1 SOFTWARE PROJECT MANAGEMENT	13
1.1 ETHICAL PRINCIPLES OF PROJECT MANAGEMENT	13
1.1.1 Truth	13
1.1.2 Trust	13
1.1.3 Review everything	14
1.1.4 Everything has equal importance	14
1.1.5 Doing the things right.....	14
1.2 SCOPE OF THE PROJECT.....	15
2 UML.....	17
2.1 GENERAL CONCEPTS.....	17
2.1.1 Everything is optional	17
2.1.2 Models can be incomplete.....	17
2.1.3 Possible interpretation	18
2.1.4 UML is designed to be extended.....	18
2.2 USE CASE DIAGRAMS.....	18
2.2.1 Use Cases	18
2.2.2 Actors	19
2.2.3 Use case relationships	20
2.2.4 Use case documentation	21
2.3 ACTIVITY DIAGRAMS	22
2.3.1 Initial and final node	23
2.3.2 Action states	23
2.3.3 Activity edges.....	24
2.3.4 Decisions	24
2.3.5 Partitions	25
2.4 SEQUENCE DIAGRAMS.....	25
2.4.1 Participants	26
2.4.2 Events and messages	26
2.4.3 Activation bars	27
3 DESIGNING USABILITY	28
3.1 IMPORTANCE OF USABILITY FOR MOBILE APPLICATIONS	28
3.2 SCREEN DESIGN	29
3.2.1 Screen real estate.....	29
3.2.2 Cross-Platform design.....	30
3.2.3 Resolution independent design	30
3.3 CONTENT DESIGN	30
3.3.1 Keep your text short	30
3.3.2 Legibility	31
3.3.3 Images	31
3.4 INTERNATIONALIZATION VERSUS LOCALIZATIONS	31
3.4.1 Internationalization	32

3.4.2	Language choice.....	32
3.4.3	Regional differences.....	32
4	WINDOWS STORE.....	33
4.1	WINDOWS STORE APPLICATION.....	33
4.2	DESIGN PRINCIPLES FOR WINDOWS STORE.....	33
4.2.1	Pride in craftsmanship.....	33
4.2.2	Be fast and fluid.....	34
4.2.3	Authentically digital.....	34
4.2.4	Do more with less.....	34
4.2.5	Win as one.....	34
4.3	NAVIGATION.....	34
4.3.1	Flat system.....	35
4.3.2	Hierarchical system.....	35
4.3.3	Navigation elements.....	36
4.4	COMMANDS.....	37
4.4.1	Canvas.....	37
4.4.2	Charms.....	37
4.4.3	App bar.....	38
4.5	TOUCH LANGUAGE.....	38
4.5.1	Press and hold to learn.....	38
4.5.2	Tap for primary action.....	39
4.5.3	Slide to pan.....	39
4.5.4	Swipe to select, command and move.....	39
4.5.5	Pinch and stretch for zoom.....	39
5	ASO PRINCIPLES.....	40
5.1	APP LISTING.....	40
5.1.1	Application name.....	40
5.1.2	Application description.....	40
5.2	APP GRAPHIC.....	41
5.2.1	Screenshots.....	41
5.2.2	Promotional design.....	41
5.3	RATINGS FROM USERS.....	41
5.3.1	Support.....	42
5.3.2	Keep improving your app.....	42
5.3.3	Ask for rating.....	42
II	PRACTICAL PART.....	43
6	PRODUCT BACKGROUND.....	44
6.1	BASIC ASSUMPTIONS.....	44
6.2	WINDOWS 8 ANALYSIS.....	45
6.3	COMPETITORS ANALYSIS.....	46
7	PRODUCT'S VISION OF SOLUTION.....	47
7.1	SPLASH SCREEN.....	47
7.1.1	Splash Screen wireframe.....	47

7.2	APPLICATION TILE	48
7.2.1	Splash Screen wireframe	48
7.3	HOME PAGE	49
7.3.1	Home page full view wireframe.....	50
7.3.2	Home page in fill and snapped view wireframe.....	52
7.4	SECTION PAGE	53
7.4.1	Section page full view wireframe	53
7.4.2	Section page in fill and snapped view wireframe	55
7.5	DETAIL PAGE.....	56
7.5.1	Detail page full view wireframe.....	56
7.5.2	Detail page in fill and snapped view wireframe.....	58
7.6	APP BAR	58
7.7	SEARCH AND SEARCH RESULTS	59
7.8	SHARING	60
7.9	NOTIFICATION MESSAGES.....	60
7.9.1	Information messages.....	61
7.9.2	Error messages	61
8	PRODUCT'S DESCRIPTION IN UML	62
8.1	USE CASE DIAGRAM	62
8.2	USE CASE SCENARIOS	63
8.2.1	Search scenario.....	63
8.2.2	View wallpaper detail page scenario.....	64
8.2.3	Set wallpaper as account picture scenario.....	64
8.2.4	Set wallpaper as lock screen scenario	65
8.2.5	Set wallpaper as desktop wallpaper	66
8.2.6	Save wallpaper	66
8.3	ACTIVITY DIAGRAM	67
8.4	SEQUENCE DIAGRAM	68
9	PRODUCT'S RISK ANALYSIS.....	69
9.1	USERS WILL NOT FIND OUR PRODUCT	69
9.2	THERE WILL BE TOO MANY COMPETITORS	70
9.3	BAD RATING FROM USERS	70
9.4	POSSIBLE VIOLATION OF TRADEMARKS OR POLICIES	71
9.5	CHANGE IN WINDOWS STORE POLICIES	71
10	USED METHODOLOGY AND SPECIFICS.....	72
10.1	SPECIFICS OF MOBILE APPLICATION.....	72
10.1.1	Simplicity	72
10.1.2	Designing for different screens	72
10.1.3	Process of approving applications.....	73
10.1.4	Personal data of users	73
10.1.5	Payments	74
10.2	USED METHODOLOGY AND ELEMENTS.....	74
10.2.1	Benefits of used methods	74
10.2.2	Analysis of used elements	75

CONCLUSION	76
CONCLUSION IN CZECH.....	78
BIBLIOGRAPHY	80
LIST OF SYMBOLS AND ABBREVIATIONS	81
LIST OF FIGURES	82
LIST OF TABLES	84

INTRODUCTION

Software applications are used in the daily life of people all over the world, millions of companies are developing their applications, and new companies are growing every day. Competition between the world's top technology companies like Apple, Google and Microsoft usually fill the headlines of the world media.

Apple, which is considered as a world innovative leader, was amongst the first companies to bring the idea of having one place for distributing all applications to life. Over time the app store became one of the most important projects for the entire Apple company. Google and Microsoft soon followed and opened their own stores Google Play and Windows Store. Application marketplaces represent one place where end users can browse through separate categories and download or purchase various applications. Now app stores have become one of the most important ways of distributing applications, but when it comes to end users the most important thing is which application is the best.

Millions of applications are being distributed from all application marketplaces. Every application contains a rating, which represents objective quality measurement. Applications developed by small, innovative way of thinking teams can achieve short term success, but for global and long term success companies need to do more. Project management is one the most important things that can help teams build good applications and create product specifications.

In the theoretical part of this diploma thesis is the description of project management tools, which are used in the process of developing software applications. The practical part contains particular product specifications of applications designed in the modern UI for distribution in the Windows store.

I. THEORETICAL PART

1 SOFTWARE PROJECT MANAGEMENT

Software project management is a new modern discipline which describes the art of leading software development. The rules of project management are universal and cannot be changed. [1] Inner processes containing project management must always reflect different company environment, its goals, possibilities and strategy. This is one of the reasons why project management can differ between starting companies with small teams located in one place and international companies with thousands of employees in different locations; however there is a sequence of general steps which should be met by every company: creating a vision, preparing specifications, creating schedules, programming and testing.

1.1 Ethical principles of project management

Project management is not only a technical skill, but it is based on ethical rules and principles which are rooted from one's moral values. Good project management should help to establish a clear and transparent company environment filled with trust, honesty and open-minded thinking. [1] Such environment in daily cooperation between programmers, graphic designers and company management should bring benefits not only for company business but it also effect daily life and improves personality of every employee.

1.1.1 Truth

In a full life cycle of software development thousands of decisions are made; from the beginning of possible business opportunities to the color background for tiniest button on the setting screen, all of these decisions have a strong effect on the final product shape. Creating a transparent environment with public information should help to make the right decisions according to truth and fact. Only sharing the entire truth and detailed information with everyone involved in the process will allow us to find the best solutions from the best qualified team members. [1]

1.1.2 Trust

Usually people from all company departments are involved in the process of software development. This type of involvement throughout a company leads to a wide variety of expertise that is vital for different decisions. The most effective way of delivering the best

product is to rely on the expertise of every team member. [1] The project manager, who usually makes the final decision, should create an environment in which each member of the team can trust that his associates are doing their best to deliver the product according to the schedule and with the highest possible quality.

1.1.3 Review everything

Today reviews are a hot topic because all detailed reviews within the development process take quite a substantial amount of time, and many companies are under pressure to shorten all schedules. The goal of document reviews is not to make a perfect documentation but to find a defect and save time in the development process. [1] It is always better to catch a problem in documentation than release a product with bug or inappropriate behavior.

In addition, reviews can have also another positive effect, while reviewing daily work, every team member is forced to take a look from a higher perspective and evaluate the commitment of all team members. This particular form of reviewing can be essential in building team spirit and respect amongst team members.

1.1.4 Everything has equal importance

This principle comes from the previous one. There is an array of different disciplines involved in software development, which can sometimes make it very difficult to keep in mind that no single part is more important than the other. Every bad mock-up from graphic designers can ruin the application as well as every wrong line of code from programmers on the other hand every nice icon can help make the best application, as well as an application's high stability from the programmers.

One of the biggest roles of the project manager is to guarantee that every decision is made after objective evaluation of all important aspects. [1] This approach strengthens the idea that everybody is equally important and can, with equal importance, affect product.

1.1.5 Doing the things right

All mentioned principles leads us to one universal statement which should be held for the whole development process, doing things correctly for first time is always better than doing them wrong and going back to fix them. [1]

1.2 Scope of the project

When the project starts the first and one of the most important things is to create a detailed scope of the project. The purpose of the scope document is to ensure that all team members are working in right direction and trying to avoid wasting time working on things which won't be used. [1]

In the beginning of the project, the project manager, who is responsible for creating the scope, should talk to most important person called the Main Stakeholder. The Main Stakeholder is the most impacted by the new project and is usually the person who came out with the idea first. The Main Stakeholder should be able to define the main goal of the new project and provide us with a list of key features. After the main idea is clear, we should be able to create a list of project stakeholders. Stakeholders is the term used to describe every person which will be involved in development of new project or will be affected by the new project after it will be released. A group of stakeholders should contain people from all possible fields of expertise. Only a wide variety of views will allow us to examine all possible risks and define all needed features and their priorities.

The scope of the project contains two parts: First is summarizing general ideas and goals which should also contain project opportunities. Second, is visions of solutions, this should be more specific and focus on describing features and release phases. [1]

Table 1. Scope of project content

1	Problem statement
1.1	Project background
1.2	List of stakeholders
1.3	Target users
1.4	Risks
1.5	Marketing or business opportunities
2	Vision of solution
2.1	List of features
2.2	List of features that will not be implemented
2.3	Release schedule

Project background should explain the reasons why the project should be started. It can also give us a global explanation which shows how problems should be solved by this project or describe the history of these problems in the company. [1] This part can also briefly describe other applications or activities which were already completed in this issue.

List of stakeholders must contain all project stakeholders with their positions, connections with the project and contact information, at least an email address. [1]

Target users similar to the list of stakeholders, this part should contain all possible end users which can be affected by project. Target users should not be necessarily described one by one, but by specific groups. [1] Each specific group should contain detailed information on the impact the end result of the project has on these special users.

Risks describe potential problems which can occur in the development process. Every problem must be given its potential impact, at least in levels easy, medium or high. [1] Action steps which can eliminate or minimize effect of every risk must also be stated.

Marketing or business opportunities are a very important part of the project scope. They are mostly the responsibility of the marketing or business departments, but if the project is a completely new application, marketing and business opportunities are one of the most crucial parts. A description of the possible benefits for business or marketing must also be stated in the scope of the project, because every key decision about planned features or graphic design must be taken into consideration with these opportunities.

List of features is probably the most important part for the development department. Every application can be described by a list of key features the length of this list depends on the detailed descriptions for the features. Every feature should contain its name and a short description of functionality. [1] In case of a complicated application, every feature can be extended with its priority, stakeholders who propose this feature, or similar previously used solutions.

List of features that will not be implemented provides a simple list of features with reasons why they will not be implemented. [1]

Release schedule shows its benefits for larger and complicated software application. This is helpful because it allows us to release the first version of application in a short time, because releasing an application with all its features is unrealistic in a short time. Every release is described by date and list of implemented features. [1]

2 UML

UML is a modeling language designed for use in the software development process. The main purpose of UML is to help software engineers describe their ideas and thoughts about what the software should look like. [2] Since we consider UML as a language, we must notice that there are strict rules on describing how pictures, diagrams or codes have to be connected.

In addition, modeling gives us the possibility to describe the whole system with every necessary detail and its complexity in a simple way. This is because when we are creating an abstract model we can simply skip parts which can be too complicated, confusing, or hard to describe. As a result, engineers are able to create simple models of applications and prepare comprehensible presentations of the whole idea to the team.

2.1 General concepts

UML has as other languages and few concepts which every engineer should reflect when describing project.

2.1.1 Everything is optional

UML is designed as a fully optional language, so we can use only the parts in which we want to use. This means that we can design easy to understand diagrams with only necessary elements. [3] Such diagrams are easy to present which minimizes time for their study by developers and graphic designers. On the other hand, this approach requires experience from engineers to ensure that diagrams will fully describe planned behavior using only a few elements.

2.1.2 Models can be incomplete

Because of optionality it is in fact very difficult to create an absolute model. In daily praxis, engineers are using as much detailed model as it is required. This happens usually due to time pressure. It will be inefficient and time consuming to prepare an absolute model for one week when programmers can write such codes in one day. Models are created not to be fully completed but to effectively describe ideas and communicate important information to the whole team. [3]

2.1.3 Possible interpretation

In the specifications of UML are described elements and the relations between them, but every company can create their own rules about how each relation can be interpreted, which can really speed up the process. Some companies may interpret composition as a C++ reference and similar rules for other relations. [3]

2.1.4 UML is designed to be extended

Especially for completely new applications which contain a desktop application, mobile versions and web presentations, must be created by a complex model including some financial or business models. Even when it is part of the UML, engineers can use profiles for J2EE and database modeling. [3]

2.2 Use Case Diagrams

Use case diagrams are one of the basic elements of UML. In some cases we can use only use cases diagrams, especially for smaller project, because they provide a quick but effective way of describing software. Use cases describe general requirements that we have on system behavior. [2] In addition every use case can be specified by use case scenarios describing every, single functionality.

2.2.1 Use Cases

Use cases are shown as an oval with a short sentence or a few words which describes the functionality. [2]



Figure 1. Use Case

Use case is from the user point of view which is based on the system they are using. It can be a simple action such as logging in but can also be of complex usage like buying an item. [3] Use case can be described also as a requirement, but then we must specifically define criteria when they are fulfilled.

2.2.2 Actors

Actors represent users in situations while they are in active contact with the system. What is important is that the user is not necessarily equal to the actor. One user can be described by many actors and vice-versa. Moreover, actors can represent not only humans but also other systems, like separate software that interacts with our system. [3] Actors in diagrams are displayed as a figure with the name as shown on Figure 1.

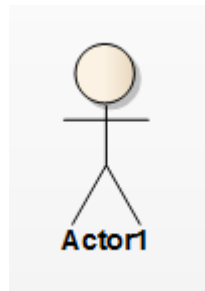


Figure 2. An actor

Actors with use cases are connected via associations. Each association can also have direction, but this direction describes only which element initiates the activity, nothing more. In using Use case for large systems we can also display diagrams with boundaries. Boundaries are designed to clearly show the specific parts of the application such as my account, orders, payments, and tabs for the e-shop platform. [2]

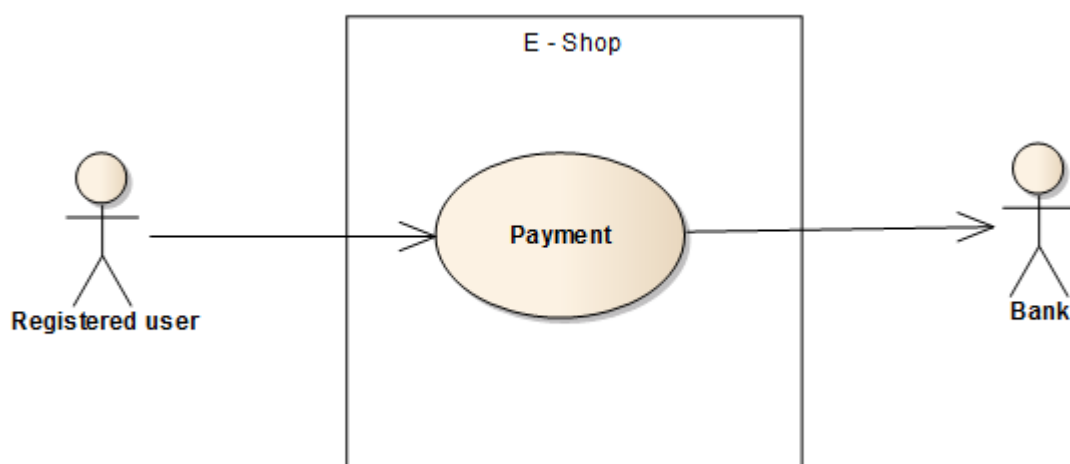


Figure 3. Human and machine actors

For example, Figure 2 describes a situation where a registered user is making a payment for order in an e-shop platform and the e-shop is sending information about this payment to the bank. In this situation we have a human and a machine actor interacting with the system. Also the direction of the association shows us that only the user is initiating the interaction when making a payment, while the e-shop never initiates a payment.

2.2.3 Use case relationships

While creating Use Case diagrams, especially for complex project, we can find out some similarities between different use cases. One use case can relate to differently to another using special relations generalization, extensions and inclusions.

Use case generalization can be used in situations when the first use case inherits the behavior of the original use case. This means that original use case can be replaced by a generalized one. Generalization is shown in the diagram with a special generalization arrow. [3]

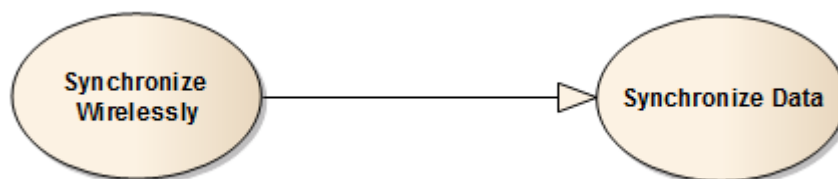


Figure 4. Use case generalization

Use case inclusion refers to the situation when one use case is including the behavior of another one. According to the rules included this can't be initiated independently, it is always used as part of a use cases sequence. Usually inclusion is used for use cases that are being used with two other use cases. Use case inclusion is shown with a dependency arrow and the word "include". [3] For example, in Figure 5 is the description of a situation where the e-shop platform is using Login as part of purchasing process and also tracking package.

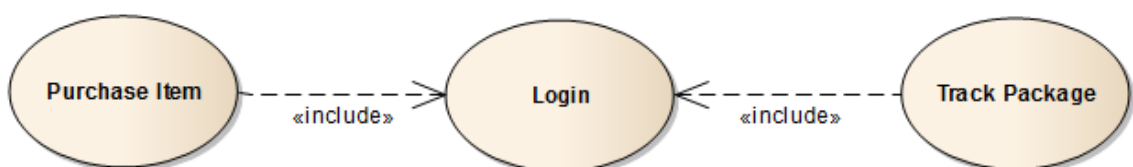


Figure 5. Use case inclusion

Use case extension is the second use case which describes the flow of events which we think is normal for first use case. Extension is usually used for use cases which are big enough to make the original use case too complicated. [3] Use case extension is shown with a dependency arrow and the word “extend”. For example, previously used e-shop platform can't purchase an item unless the item is selected.

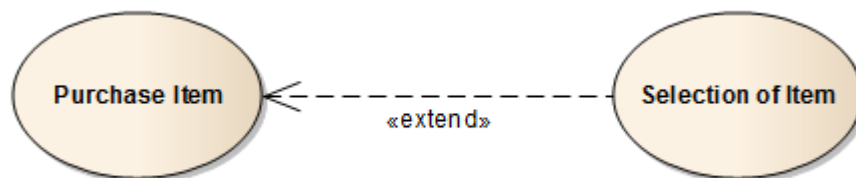


Figure 6. Use case extension

2.2.4 Use case documentation

Using a use case diagram we can provide a good description of our system, but this description will never be absolutely complete. The diagram alone will never give us the possibility to fully explain exactly how our system will achieve its goal and fulfill all criteria.[2] In fact used case documentation is not a direct part of the UML, that is why there are no strict rules for preparing use case documentation. Documentation usually contains the following parts [3]:

- **Pre-conditions:** contains conditions that must be met before use case can begin to execute.
- **End conditions:** description of status in which the system should be at the successful end of the use case.
- **Main flow:** description of steps which are executed when use case is working properly.
- **Alternative flow:** description of alternative steps for situations, such as alternative user decision or system error.

Table 2. Displays an example of use case with main and alternative flow, while also describing how a new user account is created for the e-shop platform. The Main scenario is focused on creating a new account in a standard way without any errors. In the Alternative scenario is describing a situation when the system does not verify the entered information

and the account is not created. In the alternative scenario a general error message, not one describing which special information was not verified, is used.

Table 2. Use case documentation

Step	Action
1	User initiate create a new account process
2	System shows create new account form
3	User select type of account
4	User enters account's details
5	System verifies account's details
6	New user account is created
5.1	System does not verify the account's detail
5.2	New user account is not created
5.3	System shows error message

2.3 Activity Diagrams

Use cases are used to show us what our system should do. Activity diagrams provide exact specifications on how the software will accomplish its objectives. It could be said that activity diagrams represent actions chained together to describe processes occurring in our system. [2]

A good example for using activity diagrams is describing business processes. By using activity diagram we can simply define payment processes for e-shop platforms when one step invokes another with its logical sequence.

Activity diagrams give us an opportunity to look at our model from the process point of view. The origins of activity diagrams come from flow chart diagrams. Activity diagrams are still using symbols from flow charts. Therefore activity diagrams can be used to describing processes in an easy way for most of the audience.

2.3.1 Initial and final node

To clearly show the sequence of steps that the activity diagram always needs something to start with, every activity diagram is launched by initial node. [2] In the diagram is the initial node, shown as full circle.



Figure 7. Initial node

The last element is always used as the final node which ends the activity diagram. [2] The final node is displayed as two circles with same center and different radius, where the inner circle is filled.



Figure 8. Final node

2.3.2 Action states

An activity diagram consists of several action states. Action states always represent states with entry and ending actions which leads to the next state. In the diagram are actions states represented as a rectangle with rounded sides. An action connected with this action state is described by a few words written in the rectangle. [3] Actions are in fact action steps used in process, which can be used as an example on how to verify account details. We must be careful while using words action and activity, because there is difference. While activity describes the whole process, for example creating new account, action describes only one step in this process such as verify the account details.

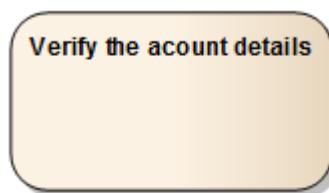


Figure 9. Action state

2.3.3 Activity edges

Activity edge is represents the transition between action states. In a diagram the activity edge is depicted as a straight line with an arrow, which shows us the direction from the first action state to the next. [3] Figure 10 shows an activity diagram for the previously described process of creating a new account, which contains a few action states and the activity edges connecting them.

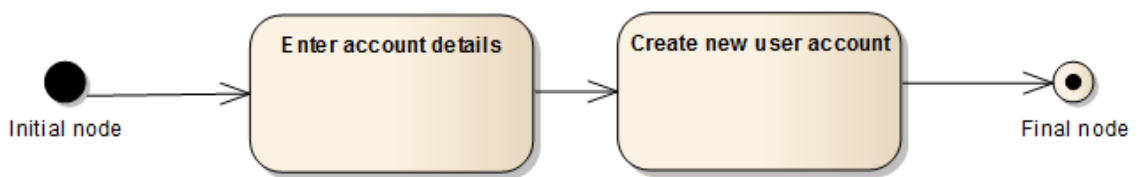


Figure 10. Activity diagram

2.3.4 Decisions

Decisions help us create the process in which we have to take into consideration the condition for different behavior. Conditions are used for representing two or more different sequences of actions depending on the used condition. [2] The Decision point is in the activity diagram shown as a diamond. The diamond is always located between the decision point and the verification results.

Figure 11 shows the decision made by the system to verify the account details, which leads to two different action states, creating a new user account or displaying an error message.

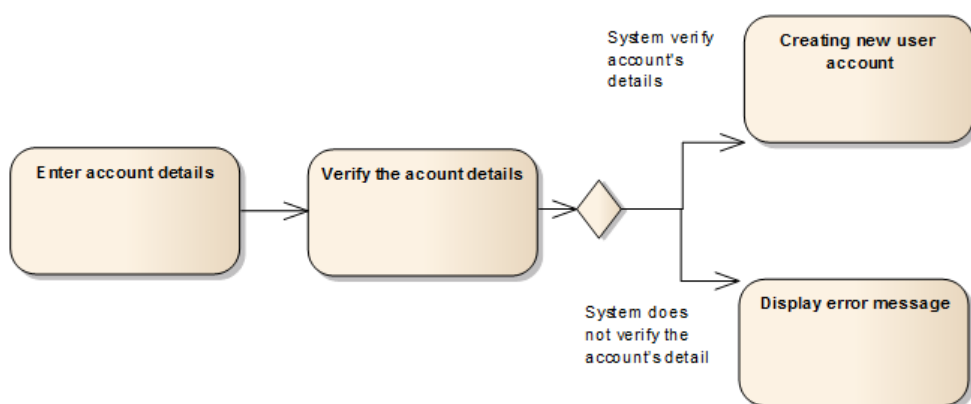


Figure 11. Decision in an activity diagram

2.3.5 Partitions

If we are creating a structural activity diagram and want to have it clear, we can use partitions. Partitions allow us to divide a diagram into areas which shows us how activities fit together and from higher perspective. Partitions can, for example, show us which company departments are responsible for which actions. [3] On a diagram partitions are shown as horizontal or vertical areas separated with lines.

Figure 12 shows a previously used activity diagram with two partitions which are divided into a user and system area, showing which actions the system and the user is responsible for.

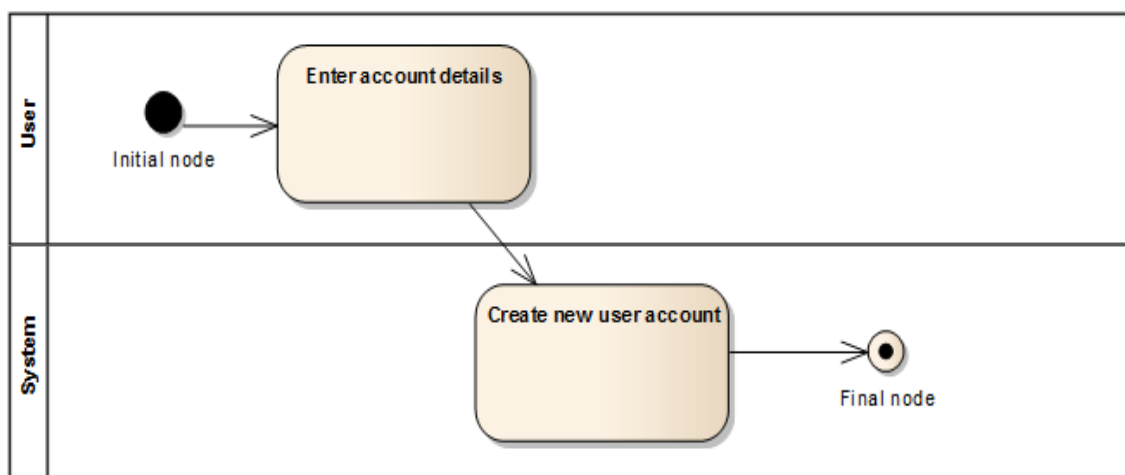


Figure 12. Partitions in activity diagram

2.4 Sequence Diagrams

With use cases and activity diagrams we have described all system requirements, all actors that affect our system and exactly how the system achieves its goals, but yet one important part remains.

Sequence diagrams help us to clarify the order of interaction. Although sequence diagrams can show us much more information, their main purpose is to show, in an effective way, the time sequence of events within an interaction. [2]

2.4.1 Participants

Participants, who represent parts of our system, play an important role in sequence diagrams. Participants interact with each other during some processes or so called sequences. [2] Participants are shown as a rectangle with the name usually aligned at the top of page and are arranged horizontally. To ensure that no two participants overlap one another, a straight line or lifeline runs down from their names. In accordance to the lifeline we can always find out which participant exists at that state of the sequence.

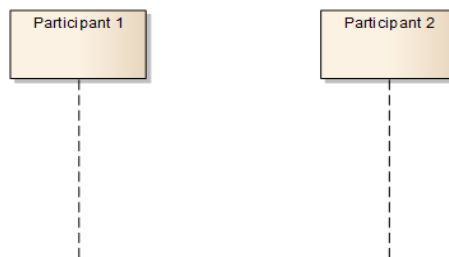


Figure 13. Participants in sequence diagram

While sequence diagrams describes the time sequence of interactions, time represents the most important factor. On the diagram, time starts at the top of page and continues down. This means that the order in which the interaction takes place from top of the diagram also represents the order in which these interactions take place in real time. [2]

2.4.2 Events and messages

An interaction is when one participant is sending message to another participant. Events represent basic elements for messages. [2] Messages are on the diagram displayed as an arrow leading from participant, which has sent a message to the other participant which accepted the message. The direction of the message can differ and it depends only on proposed application behavior.

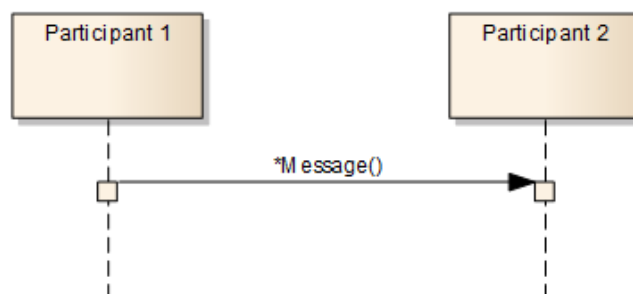


Figure 14. Event and message in sequence diagram

2.4.3 Activation bars

When a participant receives a message it forces him to invoke some action which means that the receiving participant has become active. Active participants are for a specified time busy and the length of the active time is on the diagram represented by action bars. [2] Action bars are on the diagram specified as a rectangle placed on the lifeline of the selected participant.

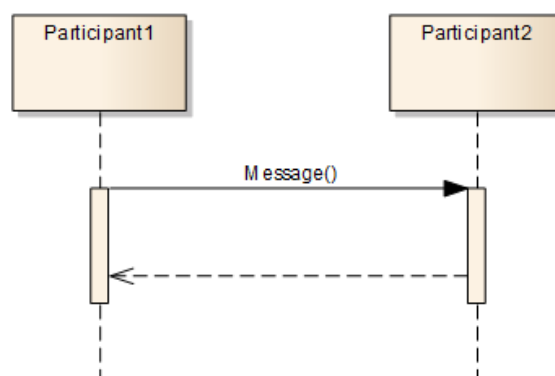


Figure 15. Activation bar in sequence diagram

Figure 15 shows activation bars representing a situation where the first participant sends a message to the second participant and is still active until he receives an answer to his message.

To fully understand how participants are busy in time, we use a combination of activation bars and different type of messages. We can use the following types of messages [3]:

- **Synchronous:** this type of message means that the sending participant is blocked and waiting until the requested operation is complete; for example, answering message arrives. Synchronous messages are in the diagram represented with filled arrow.
- **Asynchronous:** the opposite of synchronous messages, asynchronous messages are not blocked and they do not wait for the completion of the requested operation. On the diagram are asynchronous messages represented with open arrow.
- **Returning:** the last type of message is only an optional element. Returning messages can be used at the end of the activation bar and they represent messages that are returning to the original participants that sent the first message. On the diagram are the returning messages displayed with open arrow and dashed line.

3 DESIGNING USABILITY

If we want to have a complete picture about mobile applications, which are designed for distribution in application stores, we have to consider all three platforms. App stores like, Google play and Windows store, are distribute around 2 million applications, which creates an enormously competitive environment for application developers.

The end user decides which application they wish to use and best fits their needs. Why should the end user use or even pay for something that is old, slow, confusing or terribly designed especially if there are so many other options?

3.1 Importance of usability for mobile applications

If we look at software from a business point of view, we must firstly distinguish what type of software it is. For instance there will be definite differences when we are talking about desktop applications, web presentations and mobile applications.

Desktop applications are historically based on the process when the end user has purchased the software and after that are able to check the usability. [4] This process was at the beginning of the distribution evolution and is still used. For developers this provides enough time for development because after the first payment is made they have enough time to evaluate all of the end user's remarks and can make a decision on which features will be implemented. This approach is good for large complex software with history because all crucial problems were fixed and there are many satisfied end users paying for the software.

Opposite of the desktop application, web presentation has absolutely different demands on usability. With millions of web presentations we as users are well known for our impatience. If web presentations cannot present the main information quickly or the end user could not find out how to use the web they will simply leave. It can be stated that for web presentations the end user experience usability comes first and their decision to buy comes later. [4]

Mobile applications are a combination of both approaches described above. There are about 2 million mobile and rocket evolution application stores, this creates an enormously competitive environment where every single detail can be crucial for success. Like in web presentations, mobile applications rely heavily on the end users experience using the application first then whether or not if they decide to purchase the app. This is based on the

fact that every application has its own description containing text screenshots and even videos in the app store; and as a result the end user can be intrigued enough with the description that they have decided to install the application. In addition, time trial versions of mobile applications provide an opportunity to leave the application after some time due to its bad usability; on the other hand, obtaining a lifetime end user should not mean that working on the usability of our application can be delayed or halted. Many new applications are released on a daily basis in application stores, which creates an immense amount of pressure on developers. Developers must continually collect end users remarks and try to improve the usability as much as possible. In order to achieve long term success mobile applications like desktop software, require focus on every detail so they can deliver an easy to use experience and smooth usability for the end user.

3.2 Screen design

Simplicity should be the main goal while creating every screen design of a mobile application. [4] If we want to cover the whole market with our application we should also take into consideration that there are many types of hardware in which could be used by our application. We have to ensure that our application will have best possible usability for different screen resolutions. This special chapter represents the operating system. For every platform there are many versions of its OS running at the same time. For our application, this means that it should use different controls for every single OS that is necessary.

3.2.1 Screen real estate

Every single screen must have its reason, either the end user wants to see this screen or we want to present something that is important. Every screen should have an interesting content for the end user to provide them a reason to stay on it. Only 14 to 25 percent of pixels are used to display content that end user wants to see. [4] But creating a screen full of content without any whitespace would be a mistake. Whitespace is important because it guides the eye and helps the end user to group the information.

The general principle should always be that if the screen is working without a selected element, remove it. [4] After finishing the design of every screen try to imagine how this screen will work if we remove every single element from it. This practice will help us to create a simple and easy to use application.

3.2.2 Cross-Platform design

While designing the application for a special platform we know exactly if something is forbidden, which fonts are allowed, and how we can combine controls. Following guidelines is important because we should provide the end user experience that is well known for the selected platform. But if our mobile application has its version on different platforms we have to ensure the same usability for all application versions. For example, using a logo which will navigate the user to the home screen in the same area for all applications gives the user an important feeling that they are using the same application and knows how to navigate at every moment. [4] Every platform also provides an opportunity to use its own design of control elements, so we have to decide if we will use commonly used control elements typical for every platform or try to design our own control elements which will be used for all version of our application.

3.2.3 Resolution independent design

With the exception of guesswork there is no way of knowing how large of a screen our end users will have. Because of this lack of information we have to prepare special designs for all possible resolutions that will adapt to whatever size screen they will be displayed on. [4] We must also take into consideration that the end users have their own preferences. Another important step in trying to achieve the best usability will be creating customizable applications so the end user can select his own design according to his preferences.

3.3 Content design

Absolute truth is the most important content. The design should allow users to view the content in the most convenient way. The old analogy says, that when audience is leaving a theatre we want them to discuss how fantastic the play was, not how colorful the costumes were. [4]

3.3.1 Keep your text short

Research shows users read from a computer screen about 25 percent slower as they would read from a newspaper, because users do not want to read long text while using an electronic device. Due to this information we should use only about 50 percent of the text we would normally use for paper. [4] This is enhanced by the fact that we live in a digital era where everybody is carefully watching the time they spend on various activities.

3.3.2 Legibility

All of our efforts are useless when the user can not view content that motivates them to use our application. This makes legibility a priority. Described below are the rules which can help us to grant the best possible legibility [4]:

- Ensure high contrast between the text and background. With choosing different colors for the text and background on which is text displayed, we should provide optimal legibility.
- Use a background which will have minimal interference. Background patterns are easy to use and help end users to easy recognize words.
- Use big enough fonts for the text. Small fonts are not recommended, they should be used only for parts of the text which are supposed to be read only by small amount of users.
- Use static words. Dynamic effects for the presented text are recommended only for minimum cases with the goal to attract the attention of the user.

3.3.3 Images

When it comes to images we have to find a compromise between design and accessibility. For mobile applications the truth is that “An image is worth thousand words“, but we need to ensure that access to important content will not be hindered due to horribly long downloads. [4] Mobile applications commonly use cloud solutions, this is why they must communicate fast so the user does not have to wait for the downloading content. When downloading large amounts of data, servers tend to demand much a battery’s usage which could in turn have a negative effect on its usability.

3.4 Internationalization versus localizations

The possibility of distributing our application through an application store with one click to the world creates a new challenge for every developer. Users from different countries can sometimes have different mentalities, meaning they could all possibly use our application in different ways or at the very least try. This interesting situation forces us to create applications with an international usability. This is definitely not an easy issue and for some applications this can seem to be unattainable, but we must try and never give up in our quest to accomplish this goal.

3.4.1 Internationalization

To create an application which can be used by users in different countries we have to ensure that all users will fully understand every element in the application. There are also some guidelines which will be good to follow [4]:

- Do not use elements which can be offensive to any culture.
- Do not use metaphors or shortcuts which cannot be understood by some specific groups of users.
- Ensure that proper translations will be used. For example special characters (á, ã) have to be displayed properly.
- Use international time and date format. [4] By using AM/PM or the 24-hour system we are accommodating understandable time information for users from various countries.
- Try to avoid differences which can be misinterpreted. For example currency, measurement units, or abbreviations.

3.4.2 Language choice

Creating an application where all the symbols are unified without clear meaning for all users in various countries is useless, which is why we have to decide on the language that will be set as the main language for our applications. We must take this into consideration when creating a local or an international application. While releasing our application in an app store, we have to choose which countries our application will be distributed in and provide versions of the application in their language. As in desktop applications, we can also provide applications in more than one language with the possibility to select another language in the settings.

3.4.3 Regional differences

Each country can have its own specific regulations which can resolve from small changes up to completely different pricing of our application. [4] To ensure the best usability we should follow the principle that the application should always display only the parts in which can be used by user. With this approach we can prevent unwanted behavior or too many error messages which can force the user to leave our application.

4 WINDOWS STORE

Windows store is a platform designed for the distribution of applications running in Windows 8 Operating system. The release of the Windows Store platform was announced by Microsoft in September 2011 and now it has more than 60,000 active applications. [5]

4.1 Windows Store application

A Windows Store application is an application designed for use on Window 8 operating system. [5] Windows 8 is supported on both tablets and standard desktop computers; this gives us the possibility to create a completely new type of application. On the other hand this duality creates a new challenge in achieving the best UX for one application on two different devices.

One of the new elements that can help us build great UX is the support of new views which are designed to adjust to all possible positions in which the user can view the new application. [5] While designing a new application we must assume that it will be used on mobile devices, which means that users will use their fingers instead of a mouse as input. Touching input requires an absolutely newly designed set of commands which will be explained below; but one of these demands is more renowned than the rest. The Windows store application uses a new element which is well known as a tile and is replaces the icon from the previous versions of Windows operating system. [5]

4.2 Design principles for Windows Store

There are five principles which should be followed by every engineer who is designing new Windows Store application. These principles reflect basic ideas and ensure that the new application will provide the best possible UX.

4.2.1 Pride in craftsmanship

When designing an app one should focus on every possible detail to ensure that end will not encounter any unwanted behavior, because every Windows store application must be reliable and stable for its users. For high quality layout, use the new grid system, which supports good user experience. [5] Remember that creating a good application takes time and must be done correctly.

4.2.2 Be fast and fluid

Try to think about designing an application like playing chess. An application must be prepared to respond to every move by the user while also anticipating the next step. Using a touch control will help us to achieve this goal but all designers should smoothly connect all steps and reactions made by the application. [5]

4.2.3 Authentically digital

Try to use all possibilities in which the new system offers to you including new fonts, colors or typography. All together, they should help bring your application and its tiles to life. Also use new modern technologies and connect applications with clouds. [5]

4.2.4 Do more with less

Reducing your design to only essential elements helps you to create a satisfied user portfolio. Content should always be placed on the cover, so do not use redundant elements and make UI as simple as possible. [5] Users are not interested in things they did not order so do not force them to use them.

4.2.5 Win as one

Interaction with other favorite applications provides the user with the experience similar to a living organism. Take advantage of the user's own experience and allow them to use their own portfolio of favorite products by sharing content with their friend and families. [5]

4.3 Navigation

Navigation helps us to organize the content of our application so the user can navigate through it intuitively. Intuitive and minimalistic navigation helps the user to concentrate more on the content, but should also provide the answers to these three questions [4]:

- Where am I?
- Where have I been?
- Where can I go?

Navigation patterns for the Windows Store application are design to fulfill all of these requirements.

4.3.1 Flat system

Flat system is based on simplicity, but this does not mean that it is bad. Flat navigation provides us with quick movement between pages on the same hierarchical structure. The basic presumption for using the flat system is dividing the content of the application into different pages without hierarchical structure. [5] Many applications successfully use the flat navigation system to provide great UX.



Figure 16. Flat navigation system. Taken from: [5]

Typical behavior of the flat system places all page thumbnails at the top of the app bar which gives the user direct access to the entire content. [5] Achieving minimalistic UI will also help us by allowing us to not have to use the backspace button in case the entire content will not be accessible via the app bar. .

4.3.2 Hierarchical system

The Hierarchical system is commonly used in software development for web presentations and desktop applications. For the Windows Store application using the hierarchical system in combination with hub navigation creates a fast and fluid environment, even for applications with large content. The basis for using the hierarchical system is to divide the content into different sections in different hierarchical levels. [5]



Figure 17. Hierarchical navigation system. Taken from: [5]

Hub pages represent for the entry point into our application. Hub pages should contain interesting content for every section and always with possibility to navigate directly to each section. [5] For the previously used case of the e-shop platform, the hub page should be something similar to the main page as in web presentations so that it can display promoted items for every main category while also providing the possibility to navigate directly to every main category.

Section pages are the second level of the application's hierarchical structure. The section page can consist of individual items, and a detail page should exist for each. [5] In the e-shop, the section page can display the whole content of the selected main category, where the individual items represent the listed products.

Detail pages are the last level of the hierarchical structure. [5] Detail pages should display detailed content of the individual items such as listed products, news articles, or wallpaper.

4.3.3 Navigation elements

Choosing between flat and hierarchical systems is only the first step to achieving the best possible UX. Navigation elements bring our ideas to life and helps present them to the user. Using an inappropriate navigation element can ruin the whole navigation system. The Windows Store application can use up to ten navigation elements [5]:

1. **Header and back button:** the header describes the current page and the back button allows the user to go back to the previous page.
2. **Hub page:** as the entry point of the application, hub pages should display examples of the entire content to user so they can make a picture of they can do or view.
3. **Content sections or categories:** can help us to clearly display the content of the selected category.
4. **Semantic zoom:** the new effect of semantic zoom helps users to easily navigate between individual levels of the hierarchical structure.
5. **Top app bar:** provides direct access to individual pages.
6. **Header menu:** because the header menu is available on any page of an application, it is primarily used for quickly navigating from one section to another.
7. **Home link:** is usually displayed at the bottom of the header menu and gives us the possibility to navigate from any place in the application to the home page.
8. **Bottom app bar:** is used as a place for displaying all control commands, which can also change the current page.

9. **View, sort or filter elements:** standard elements that change the order or displayed content on the page, can also be used in desktop applications and web presentations.
10. **Edge:** swiping from the edge of the screen displays app charms.

4.4 Commands

In comparison to web presentations or standard desktop applications, the Windows Store application has more surfaces on which we can place commanding icons. Displaying the right command icon in the right place is another important step in the process of creating the perfect application.

4.4.1 Canvas

Using canvas only, end users should be able to complete crucial scenarios. [5] Placing a command on the surface which is not directly accessible means that users will have to make an additional interaction to display the appropriate icon.



Figure 18. Canvas. Taken from: [5]

4.4.2 Charms

Charms represent app commands which should be common for all applications in the Windows Store and even for the Windows 8 operating system. [5] Try to avoid duplicating functions by placing icons with similar behaviors in the app bar.



Figure 19. Charms. Taken from: [5]

Charms provides following functionalities [5]:

- **Search:** provides a search feature for the content of our application, including other applications on the operating system.
- **Share:** allow us to share the content of our application with other installed applications or users.
- **Device:** allows us to enjoy audio, video and images from our network.
- **Setting:** allows users to configure our application with UI settings common for all applications.

4.4.3 App bar

The App bar is the surface dedicated to displaying commanding icons which are relevant to the displayed content. The App bar is not directly visible, but users can display the app bar by swiping from the top or bottom edge of the screen. The App bar should be transient and disappear right after the user taps on icon or onto canvas. [5]



Figure 20. App bar. Taken from: [5]

4.5 Touch language

4.5.1 Press and hold to learn

By touching and holding displays, functions for selected individual elements such as context menu or hints, will become available. [5]



Figure 21. Press and hold interaction. Taken from: [5]

4.5.2 Tap for primary action

Tapping initiates primary action for the selected element. [5]



Figure 22. Tap interaction. Taken from: [5]

4.5.3 Slide to pan

Slide interactions are mostly used for moving, drawing or writing. [5]

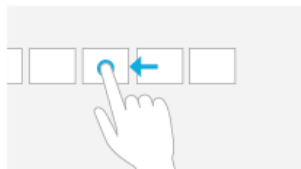


Figure 23. Slide interaction. Taken from: [5]

4.5.4 Swipe to select, command and move

Sliding one's finger a short distance around objects will select the objects and display the app bar with possible commands for them. [5]

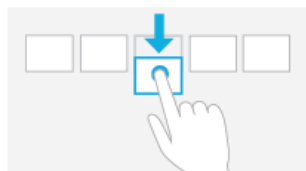


Figure 24. Swipe Interaction. Taken from: [5]

4.5.5 Pinch and stretch for zoom

Pinching and stretching interactions are commonly used for changing the size of the canvas. In addition, Windows 8 allows this interaction to navigate to different pages. [5]

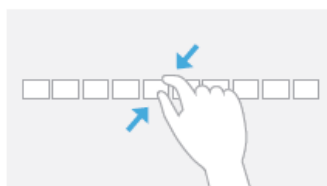


Figure 25. Pinch and stretch interaction. Taken from [5]

5 ASO PRINCIPLES

Developing new applications for application stores is a very competitive environment. The Windows Store with about 60,000 active applications is quite possibly the smallest platform, but the number of active applications here is growing rapidly every month. Developing the perfect, innovative and intuitive application is helpless if the user cannot find it. This makes Application Store Optimization so important. Based on SEO, search optimization is only one part of whole ASO.

5.1 App listing

App listing provides your app with cover and is literally sells your app. Description is first part of this and can be created independently in the development process. The sooner you start creating a description the better the description will be prepared for the release date.

5.1.1 Application name

The application name is the first thing the user sees when finding your app. Choosing a name which is competitive, creative and innovative helps users remember your application from the beginning. When choosing your application name try to follow these universal principles [5]:

- **Be creative:** try to think in a new innovative and creative way. Slang or common words may be helpful.
- **Do not use other trademarks and names:** ensure that your chosen name belongs to you and is not a registered trademark of other companies.
- **Reserve your name as soon as possible:** while registering your application name you have 12 months to release the application. In that time period no other developer can reserve or release an application with that name.

5.1.2 Application description

A good description makes an app unique so think about the description as a chance to sell the application to users. On the other hand bad description with wrong keywords can totally ruin application and make it almost invisible. Below are described universal principles that can help with creating unique description [6]:

- **First two sentences:** are probably most important, because if they do not attract the user's attention they will probably leave.
- **Use a proper length:** a good description is not too short, but also not too long. The length depends on the app content and its features, but generally a good description should contain between 200 and 3000 words.
- **Include a features list:** using a bulleted list for features is a good idea, because it is clearly describes precisely what your application is designed for.
- **Use catchy language:** try to avoid dry language, make your description a living organism with catchy phrases.
- **Keywords:** at the very beginning define keywords that will describe your application for the search engine. From the user's point of view, keywords make a list of words the user will probably use to help find your application.

5.2 App graphic

5.2.1 Screenshots

For screenshots use images of your application which describe its key features. The screenshot image should be at least 1366 x 768 or 768 x 1366 pixels and should provide the possibility to use higher quality screenshots. [5] In time your application will definitely change, maybe the whole design will change, so do not forget to update screenshots of your application in the store.

5.2.2 Promotional design

While screenshot promotional images provide space for graphic designers to showcase their skills and expertise, promotional images help to create a strong brand for its application. Use images that feature logos, famous characters, or other branding elements that help define the application. [5]

5.3 Ratings from users

Rating allows users to show how satisfied they are with using the application. Ratings are typical for application stores and make a big difference according to the standard desktop applications or web presentations. Ratings not only provide quick feedback for new applications, they also represent the viral environment for store applications that reflect the needs and wishes from customers. [7]

Ratings from all users are visible to the public, which makes it an important criterion for other users thinking about installing your application. As quickly as a good rating can boost your application, a negative reaction can completely destroy all of your work, so be really be aware of your user's need.

5.3.1 Support

Providing contact email for support gives users the possibility to contact you directly. This ensures the user has a direct line to you and can make you aware of the problems they incur while using your application, this could in turn lower the chances of them giving your application a negative rating in the application store. [7]

5.3.2 Keep improving your app

For unsatisfied users getting a reaction quickly is crucial. Reply to all users emails as quickly as possible, even if there is no quick solution to the described problem. Replying to emails is only the first step, you must periodically check all remarks and decide which proposed improvements will be helpful and implement them into the application. An excellent way of get a positive rating is asking the user to write a review after their problems were solved.

5.3.3 Ask for rating

Asking for a review inside the application can be tricky, because any unwanted or disturbing behavior can lead to a negative rating from the user so it must be done in the appropriate time and place. The Basic principle is to not be too offensive but think about how the users are using your application and ask them for a review only when they could have a reason to be satisfied. [7] This can be after the user passes some levels in the game, after a few times when users launches your application, or when the users are sharing the content from your application on social networks. Giving the user the possibility of reviewing the app at a later time is can also be important.

II. PRACTICAL PART

6 PRODUCT BACKGROUND

The first step for making a product should be to fully comprehend the main goal of the new application, to understand the origins and needs of the stakeholders, and their reasons for developing the new product. These statements should not only be described in a document, they should also be integrated in every meeting, decision, and line of code.

6.1 Basic assumptions

The new operating system, Windows 8, provides its users with plenty of new features and brings to life a completely new UI. It is well known that Microsoft is trying to focus on two absolutely different devices with one version of the operating system and that seems to have its pros and cons. But the fact that every new design will make one group of end users unsatisfied is indisputable. Users which were content with the previous UI and design used on Windows 7 will not leave their old habits behind easily. This is exactly the space needed for any application that can help users with the customization of a new operating system.

Our new product will be a modern application designed for distribution in the Windows Store. The application's main purpose is to provide users with high quality wallpapers which can help them adjust Windows 8 to their needs. Below is described the main ideas for the application.

- **Free application:** an application designed as a free product. The main purpose of the application is to provide users with customization possibilities and help them use the new operating system quickly. From a business standpoint this application is aimed at building the company brand from the positive reactions or viral distribution from end users.
- **Wallpapers divided into categories:** used wallpapers will be divided into categories which help users navigate easily through whole content of the application. Users are likely familiar with categorization from using similar web presentations, so it is almost like routine behavior.
- **Search feature:** wallpapers will contain a name which allows the user to search through our application and find their favorite wallpapers easily. Search features should be based on the Windows Store guidelines represented with charm

commands. Utilizing system elements will strengthen the use of charm commands making them common behavior.

- **Set wallpaper as lock screen:** a function available for every wallpaper which allows the user to set the selected wallpaper as a lock screen image.
- **Set wallpaper as profile picture:** a function available for every wallpaper which allows the user to set the selected wallpaper as a profile picture.
- **Set wallpaper as desktop wallpaper:** a function available for every wallpaper which allows the user to set the selected wallpaper as the desktop wallpaper.
- **Save wallpaper:** a function made available for wallpapers that allow the user to save the selected wallpaper as a file.
- **Share wallpaper:** a function available for all wallpapers which allow the user to share the selected wallpaper on social networks with friends.

Possible improvements for future versions of an application are:

- **Screensavers:** add packages of predefined screensavers which can be installed in ways similar to wallpapers.
- **Cursors:** add packages of predefined cursors which can be installed in ways similar to wallpapers.
- **Themes:** add packages of whole customization themes which can be installed in similar ways to wallpapers. Because themes change more elements in the operating system, they should be presented with a different logic.

6.2 Windows 8 analysis

Windows 8 provides some places for customizations, some of them are well known from the previous versions of Windows operating system and some are new. The first customization can be done by users on the Start screen, there users are allowed to change the background and color. The background is displayed only on the Start screen but changing the color will have an impact on other surfaces in the system. Another customizable part is the Lock screen, which is displayed when system is locked or launched from sleep mode. On Lock screen, users can change the background picture and add new pictures representing every created account. Common customization represents the desktop background image. Using their hand-selected wallpapers for the desktop background, users can change their device into a personal friend.

6.3 Competitors analysis

Background wallpapers HD provides square and wide main application tiles with 4 wallpapers which clearly describe the application. After the first launch of the application tile, change to live and display the wallpaper or simple text on a dark background. Loading the application for the first time with the text explanation will take a substantial amount of time. The main features are set as lock screen, the account picture, or saved to pictures. On the application's main screen the settings are also displayed. Tiles for the settings can be an interesting design element but they are a redundant element for UI.

Bing wallpapers provide simple square static tiles, which in comparison to other competitors, is its first weakness. Loading the application for the first time was similar to the others but there was no notification text, only a simple progress bar. The wallpapers are on the hub page divided into two categories, "Recents" and "Promotional pictures". The main features for all wallpapers are set as lock screen, download, download onto phone, and share. The App does not allow users to set as an account picture or desktop wallpaper.

Cars HD provides static wide application tiles with car wallpaper, which means no interaction but simple solutions that clearly describes the application's purpose. Initial loading time is quick, but the launch screen with only one simple colored background and centrally located wallpaper looks unprofessional. The application only contains several wallpapers of cars displayed all on the main page. Tapping on the wallpaper initiates a slideshow of all the wallpapers with possibility to save and set as them as the lock screen. With a minimum of features, this application is definitely simple and is probably designed for a specific group of users.

Daily wallpapers provide live, static, square and wide tiles with the design and application names which are simple but pretty recognizable between competitors. Daily wallpapers are a free application with displayed advertisements and possible upgrades to paid accounts without ads. The main page contains the following sections: today's wall, trending images, local images, camera, Skydrive and Facebook. Through this element the app allows users to directly access images located on Skydrive and Facebook, resulting in the most sophisticated application from chosen its competitors. There are some missing features for setting the account picture and directly setting the desktop wallpaper, however all wallpapers can be saved, set as the lock screen and shared on social media which results in daily wallpapers being the clear winner amongst its competitors.

7 PRODUCT'S VISION OF SOLUTION

Vision of solution provides the full description of the functionalities using wireframes and UML elements. It is used as documentation for programmers and graphic designers for developing proper functionality.

7.1 Splash Screen

The Splash screen is displayed after starting the device while the application is loading. It is important that the splash screen can be displayed for a longer time while the application encounters any problems with its server connection. In the case this was to happen, a system progress bar appears.

7.1.1 Splash Screen wireframe

Splash screens consist of a background and a main image. The Main image described in the wireframe will display the application's name and several previews of wallpapers arranged randomly in a pile. The Main image will be displayed in the center of the screen in landscape and portrait views. The Background for the splash screen will be approved from proposed mockups from the graphic designer, but it should consist of one color without any effects.

The Progress bar, if necessary, will be displayed below the main image so it will be clearly visible.

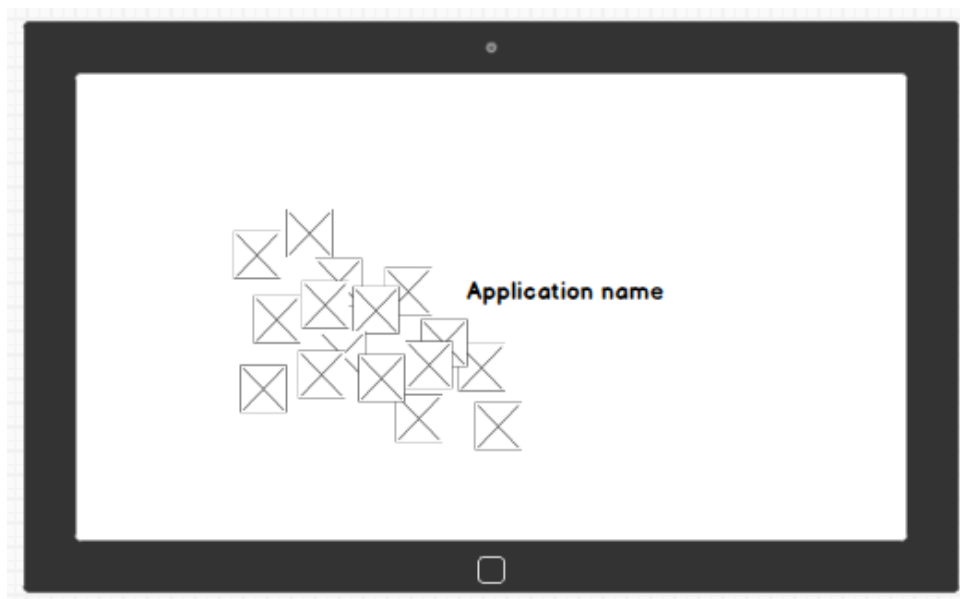


Figure 26. Splash screen wireframe

The Splash screen must be prepared for all possible scales of the screens described in table 3.

Table 3. Splash screen sizes

Scale	Image size [pixels]
1x	620 x 300
1.4x	868 x 420
1.8x	1116 x 540

7.2 Application tile

Every application on the start screen is represented with its tile immediately after successful installation. The Start screen allows us to use 2 different types of tiles; the user can switch from a wide to a square tile if the application provides both types. Another possibility is to prepare a live tile to display content from the application. We will use both wide and square tiles and combine them with live tiles.

7.2.1 Splash Screen wireframe

Static tiles will be prepared in square and wide sizes. Both types should contain the application logo and name according to Figure 28.

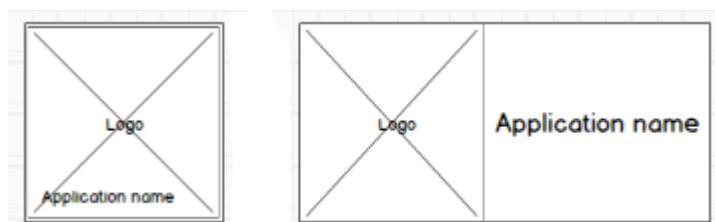


Figure 27. Square and wide static tile

Live tiles will also be prepared in both square and wide sizes. Both types will change the displayed preview wallpaper every 5 seconds. Wallpapers from the categories will be alphabetically listed during the preview.

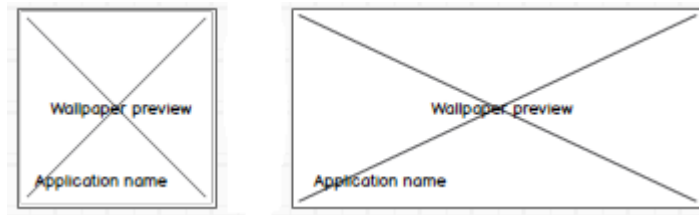


Figure 28. Square and wide live tile

For the splash screen, application tiles must be prepared for all possible scales of screen according to table 4.

Table 4. Application tile sizes

Scale	Square tile size [pixels]	Wide tile size [pixels]
0.8 x	120 x 120	248 x 120
1 x	150 x 150	310 x 150
1.4 x	210 x 210	434 x 210
1.8 x	270 x 270	558 x 270

7.3 Home page

The Home page is displayed immediately after the Splash Screen, or immediately after the entire content of the application has loaded. Our application is designed in the hierarchical structure and the home page represents the hub page for the navigation system.

The Home page's main purpose is to display all the main categories and give the user the possibility to navigate through the section page for a selected category. In addition, every main category and some wallpaper, via the wallpaper preview, should be visible with the user being able to directly navigate through the detail page to select their preferred wallpaper.

While we are considering the main page as the hub page for the whole application, the main page will not be visible with the back button, because there is no page which is higher in the hierarchical structure.

7.3.1 Home page full view wireframe

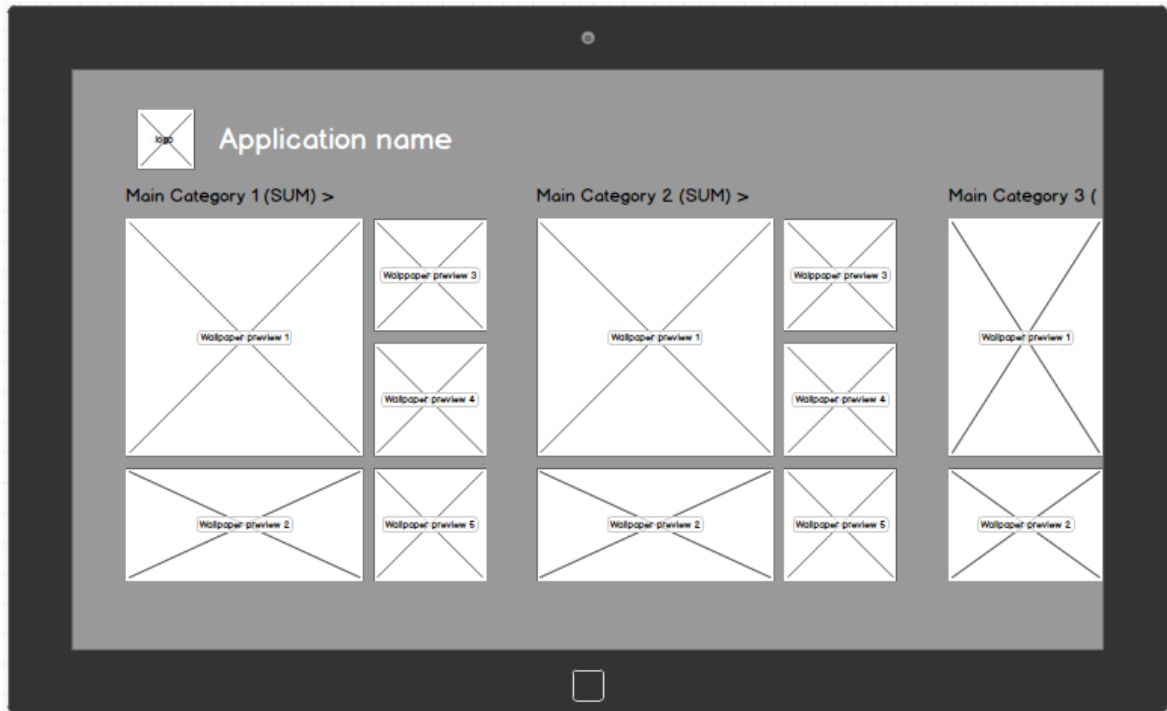


Figure 29. Main page full landscape view wireframe

The entire application, starting with main page, will be based on the grid system designed for organizing elements. The following elements will be displayed on the main page:

- **Background:** uses one color without any additional effect. The Background color will also be the same for the section page and it will be approved together as part of the color scheme for the whole application.
- **Application name:** represented by Static Text.
- **Logo:** an image representing the application created by graphic designers.
- **Main category:** each main category will be presented as described in wireframe. Categories will be displayed in alphabetical order and categories which will not be visible should be displayed with a common sliding functionality to the right side.
 - **Main category section label:** consist of the main category name and (SUM) representing the total number of wallpapers in each category. Tap on the main category section and the label opens the section page for the selected category.
 - **Wallpaper preview 1 – 5:** previews of the first 5 alphabetically ordered wallpapers for every category. In case the category will contain 4 or less wallpapers, other previews will not be presented. Tapping on the wallpaper preview opens the detail page for the selected wallpaper.

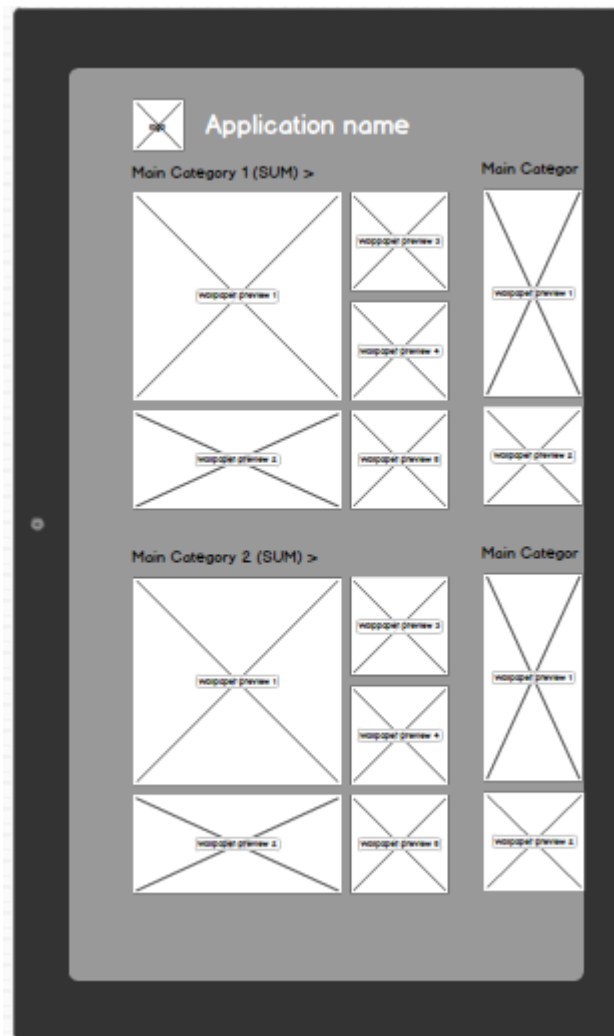


Figure 30. Main page full portrait view wireframe

Although portrait view has a minimum effect for standard desktop computers running windows 8 operating system, we will proceed implementing this function. The use of portrait view is common behavior for both tablet and mobile devices. The duality between desktops and tablets cause many problems for developers, but for users the functionality of the application displayed in landscape and portrait views are of the most basic of features.

The content in our application will be displayed in the same way it is displayed in landscape view but adjusted for the portrait view. The main difference is that not 1 but 2 categories will be displayed in one column, meaning more content can be displayed on one page. In addition, content will also be accessible by sliding to the right side similar to the logic of the landscape view.

Users can change between portrait and landscape views thanks to standard tablet behavior provided by the operating system.

7.3.2 Home page in fill and snapped view wireframe

The application will use new, fill and snapped, types of view which are supported in Windows 8. Both new view types can be opened by the user on a device with Windows 8 to ensure maximum user experience by running more applications on one screen.

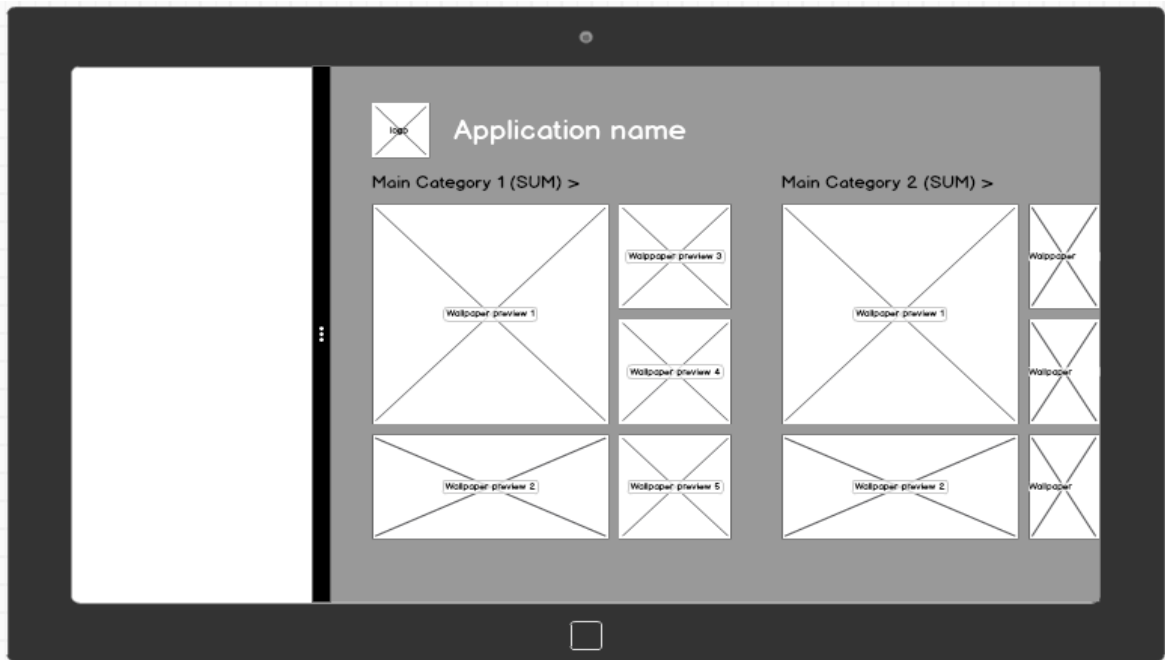


Figure 31. Main page in fill view

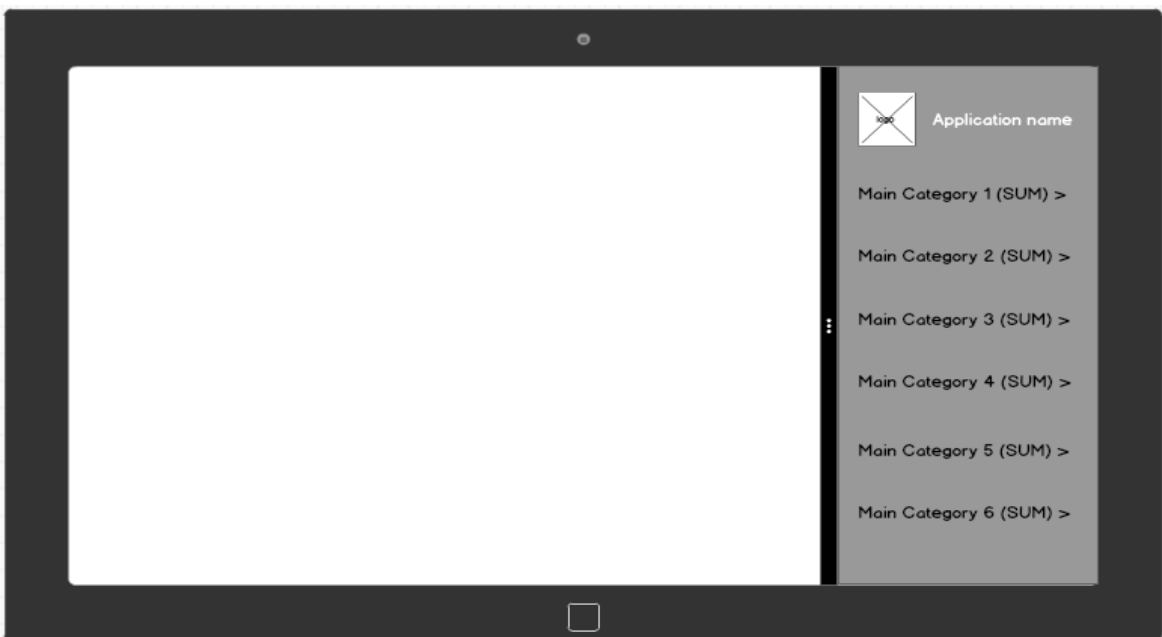


Figure 32. Main page in snapped view

While fill view has enough surface for displaying application content in standard way used in full view, purpose of snapped view is little bit different. Snapped view is not used for

displaying whole application content but it is designed to stay in touch with application while some other activity is needed. According different goals for snapped view will display list of main categories represented with its Main category section labels.

7.4 Section page

In our application The Section page will be the representing page for the selected main category. The Section page will be displayed by the system after the user taps on the Main section label on the main page and by doing so; every main category will have its own section page for displaying content. Category content in our application is represented by wallpapers. From a navigation point of view, users can navigate from the section page back to the main and detail pages for the wallpaper preview.

All section pages will be displayed in the same way which provides us with a generic approach that in turn can be extremely effective when it comes to programming. Different designs for every single section page will more than likely be user friendly but it will be extremely time consuming for implementation. Taking into consideration that the app is designed for free, different UI for section pages is excluded from the first implementation phase however it can be included in later phases if this special functionality will be requested by users.

7.4.1 Section page full view wireframe

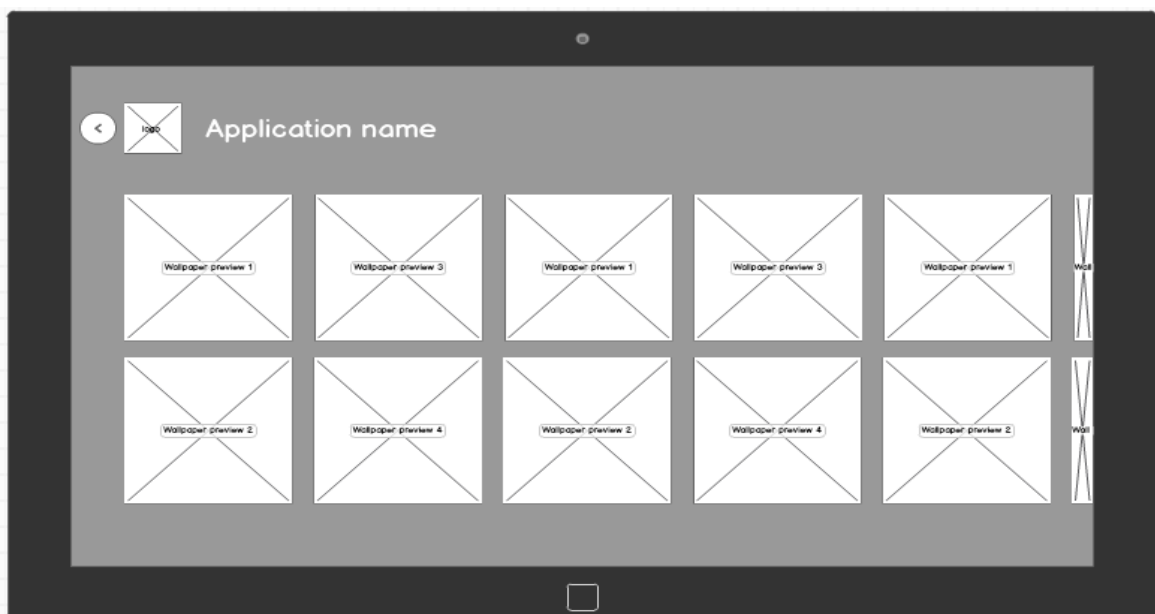


Figure 33. Section page in full view

The Section page will be as a default, displayed in full view, from which users can switch to the supported fill and snapped views. The Section page will contain the following elements:

Back button: represent a navigation element. Tapping on the back button will navigate the user back to the main page.

Wallpaper preview for the section page: represents the way in which wallpapers will be displayed on the section page. Wallpapers will be ordered alphabetically and will be displayed in 2 columns as described in wireframe. Every wallpaper preview will be presented in 225x225 pixels. This size is a compromise between displaying as much content as possible and building positive user experience with clear and visible content.

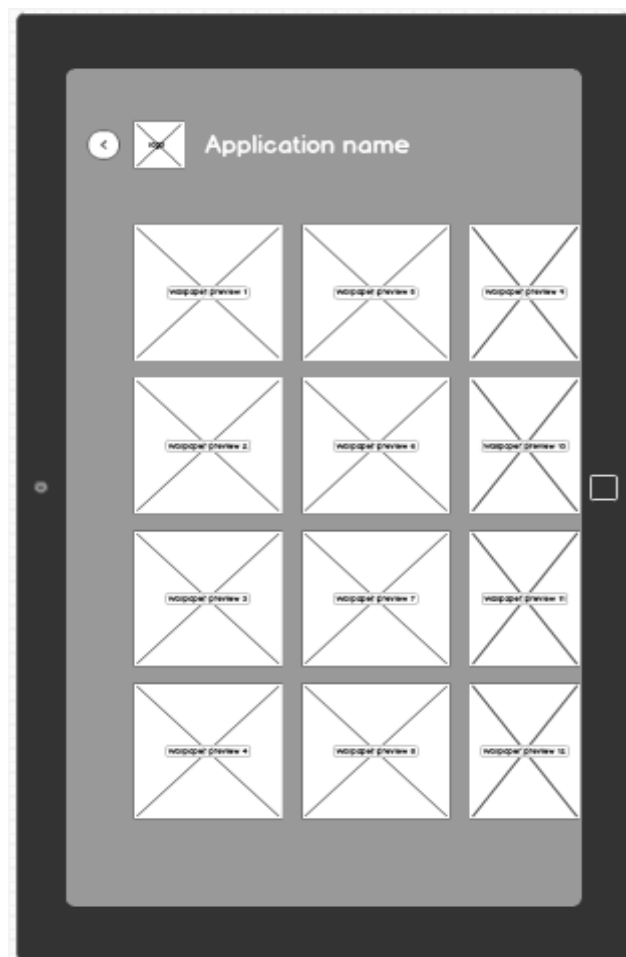


Figure 34. Section page in full portrait view

7.4.2 Section page in fill and snapped view wireframe

The section page will support showing category content with only an adjusted amount according to the fill view.

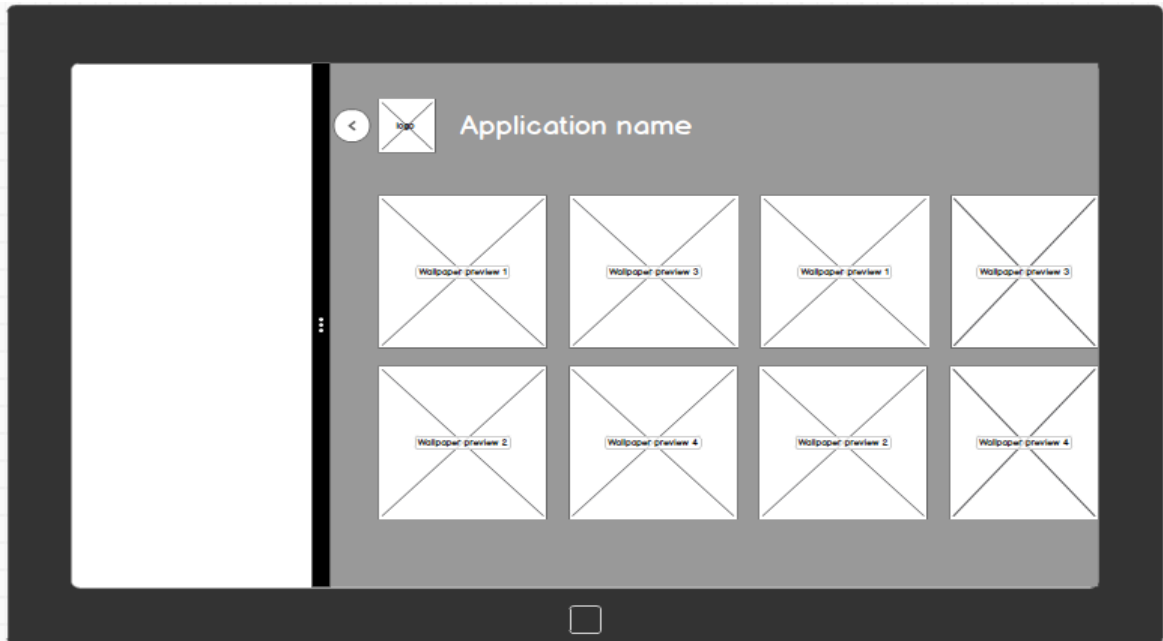


Figure 35. Section page in fill view

For the snapped view the situation is slightly different. The Snapped view will show the user the content of the selected category despite the lack of space. The vertical scroll bar, located on the right side, allows the full content to be displayed.

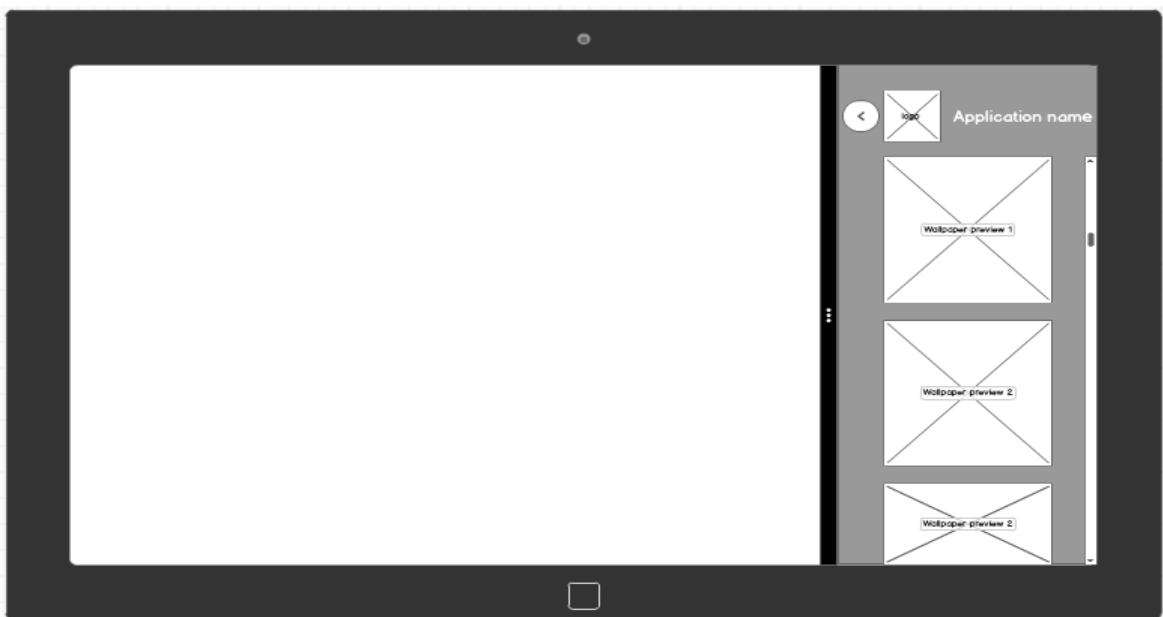


Figure 36. Section page in snapped view

7.5 Detail page

The main objective of the detail page is to present selected wallpapers in high resolution to users. The detail page is always displayed after the user taps on the wallpaper preview from the main page or the section page.

Wallpapers on the detail page will be displayed in the highest possible resolution to provide users with the highest possible user experience. With the selection of the wallpapers in the application in our hands, it ensures that all used wallpapers will meet our requirements.

The detail page is from the lowest level of the hierarchical structure. Users can display the detail page from two entry points, the main and section pages, which creates duality for navigating from the detail page. If the user opens the detail page from the main page then the back button will navigate him back to the main page. The same logic is used for the back button when the user opens the detail page from the section page. The Back button simply navigates users to the previous page.

7.5.1 Detail page full view wireframe

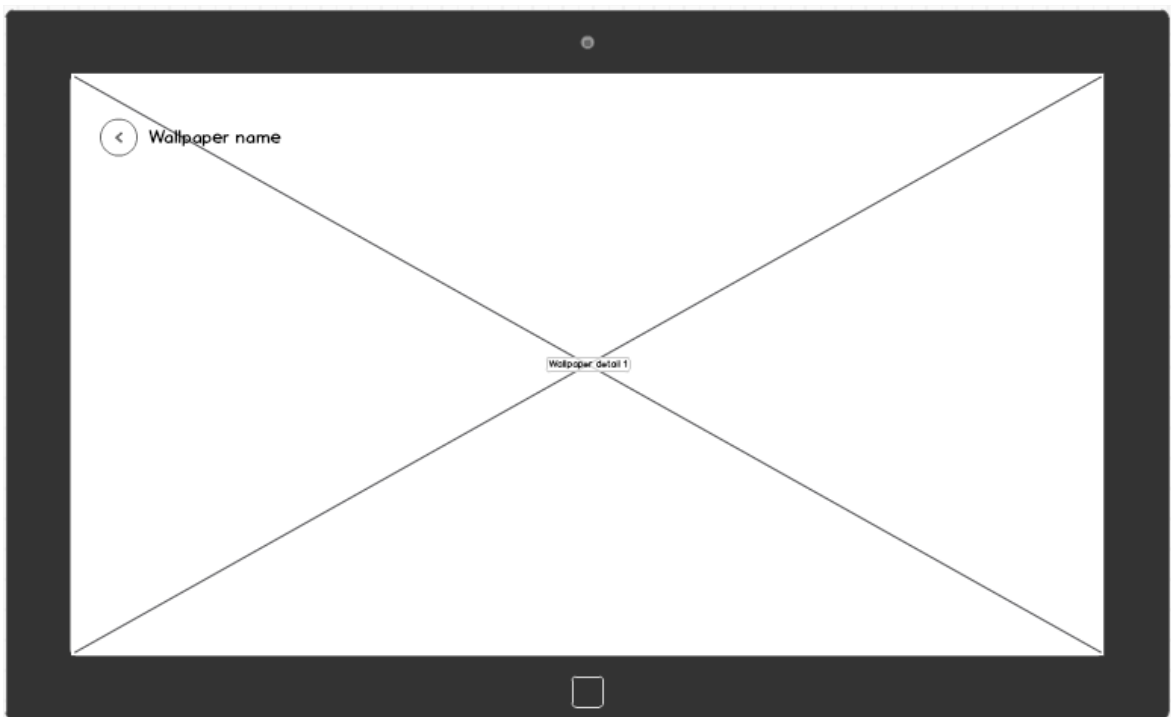


Figure37. Detail page full landscape view wireframe

The Detail page contains following elements:

- **Back button:** tapping on the back button navigates the user to the previously displayed page.
- **Wallpaper name:** represents name of the file with wallpaper. Wallpapers are inserted into the application by us meaning that we are undoubtedly responsible for checking the proper wallpaper names.
- **App bar:** a supported feature for the detail page. The App bar is initiated by standard behavior, sliding from the bottom of the edge described in the guidelines for the Windows Store application.

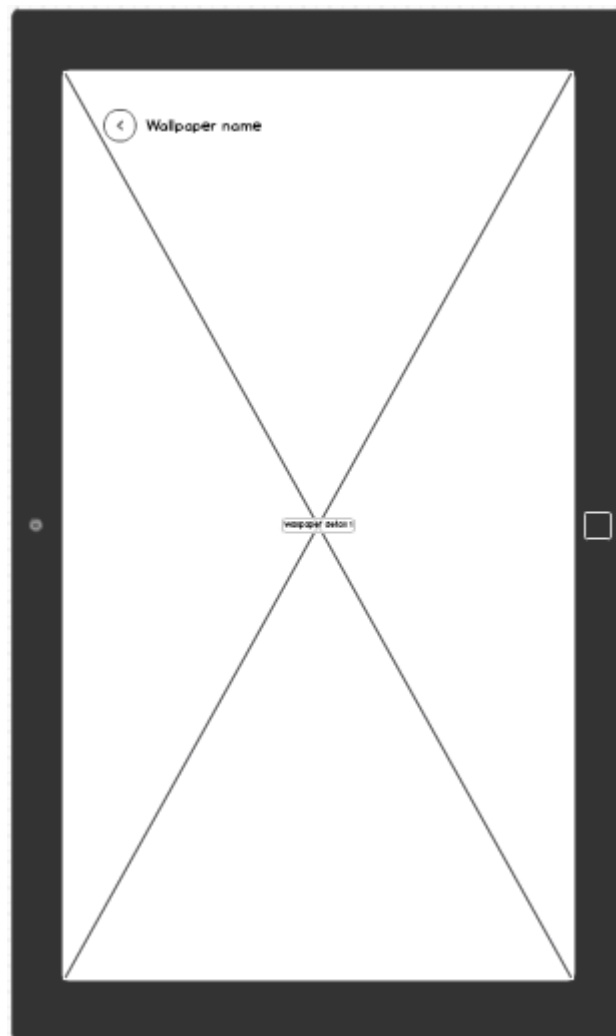


Figure38. Detail page portrait view

The detail page also supports portrait view as described in figure 38. The portrait view will also contain the exact same elements, and will manifest only when the tablet or other device is rotated.

7.5.2 Detail page in fill and snapped view wireframe

As with other pages, the detail page will also support fill and snapped views; but the detail page is limited because proper display content is needed for the whole screen. As a result when the user initiates fill or snapped views for the detailed screen application it will not only adjust to the selected type of view, but also display the section page of the category in which the detail screen list wallpaper is from. This behavior will allow us to support fill and snapped views for all application pages.

7.6 App bar

The App bar is a default system element which is described in the Windows Store guidelines. The App bar is designed to display control commands available for displayed content. The App bar is displayed always when the user slides from the edge on the bottom of the screen.

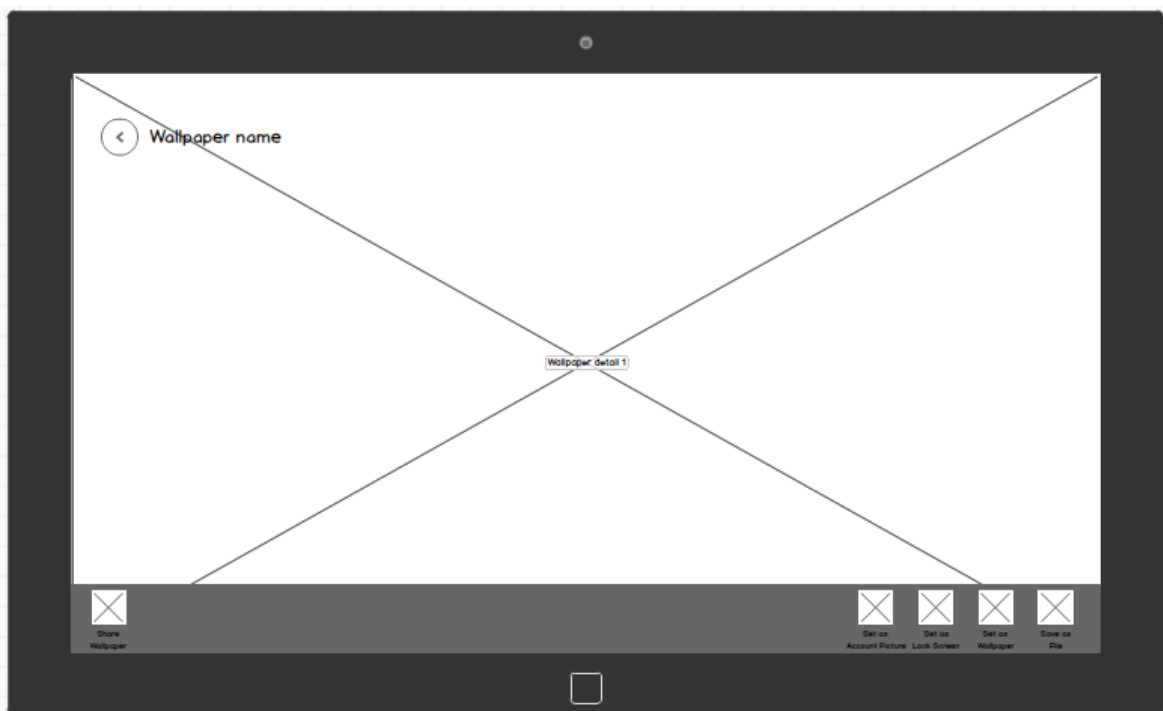


Figure39. App bar on Detail page

Our application will use the App bar only on the Detail page, because for the Main and Section pages different scenarios can be initiated directly from the canvas by tapping on the wallpaper previews. The App bar is supports both landscape and portrait views for the detail page, with both displaying the exact same elements in the same order according to the wireframe. Only on the Detail page are scenarios which can be initiated from the app

bar and these scenarios are represented by icons. One app bar will display the following icons with text description from the wireframe:

- **Share wallpaper:** initiates shared dialog with social networks for displayed wallpaper. This element is not redundant because it has different functionality, comparing and sharing icons in charms.
- **Set as account picture:** initiates the scenario in which the system will set the displayed wallpaper as the account picture.
- **Set as lock screen:** initiates the scenario in which the system will set the displayed wallpaper as the lock screen.
- **Set as wallpaper:** initiates the scenario in which the system will set the displayed wallpaper as the desktop wallpaper. The Wallpaper will be centrally aligned if it will have a different size.
- **Save wallpaper:** initiates the scenario in which the system will save the displayed wallpaper as a file. The file will be saved as a default into the user's downloads folder in the system.

7.7 Search and search results

The Search feature will be initiated with the charm search element, which is the default solution provided and recommended in the guidelines of the Window 8 operating system.

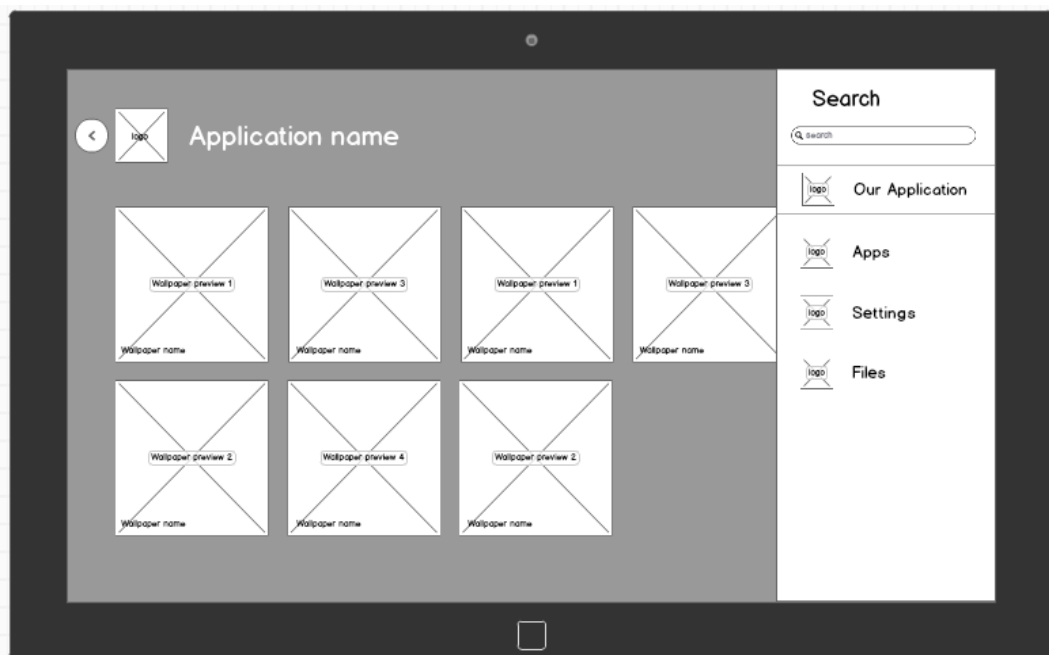


Figure40. Searching application with results

The search in the application will be provided in part because of the names of the wallpapers. The search results will be displayed as a section page with wallpapers that met the search query criteria. The search result wallpapers will be displayed in alphabetical order with one new component, the wallpaper name.

- **Wallpaper name:** text information with the wallpaper's name will be displayed in every wallpaper preview in the lower left corner.

7.8 Sharing

The Search component, displayed on the right side in figure 40, is the default system component used for sharing. Our sharing scenario will use the same system components displayed applications use for sharing. While used for sharing, this component has been already described within the search chapter; we will not add another individual wireframe for sharing.

7.9 Notification messages

Notification messages are used to present important information for the user. For notification messages we will use the system component described in the Windows Store guidelines.

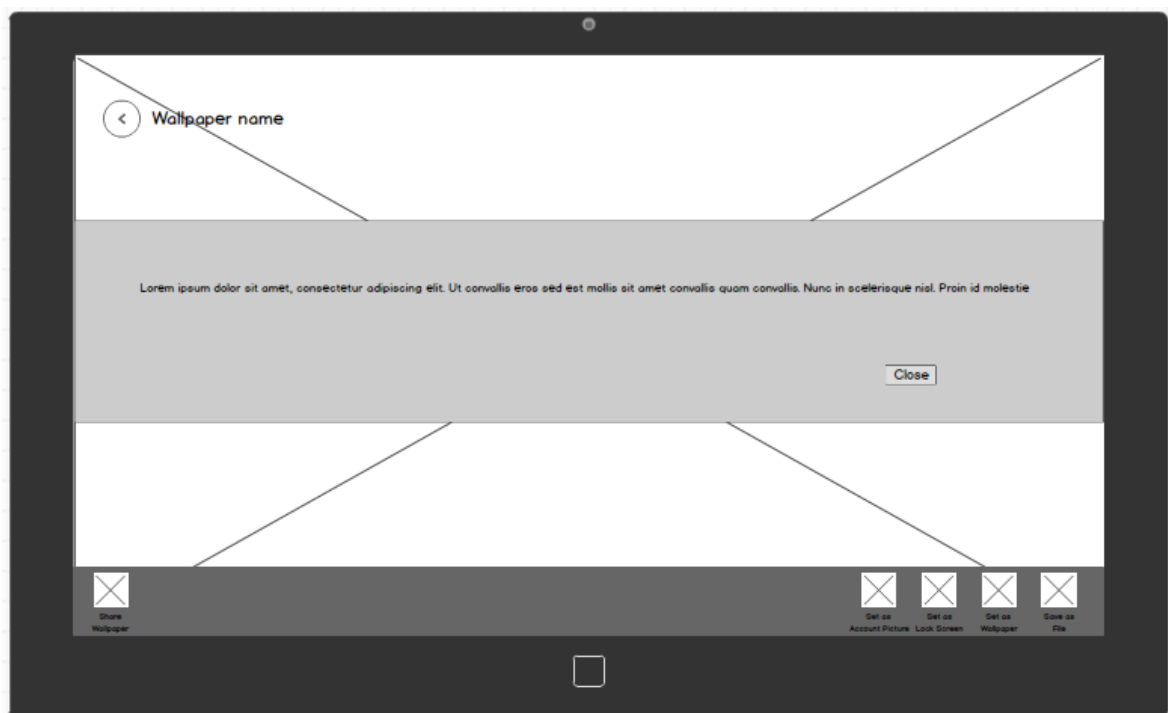


Figure41. Notification messages

7.9.1 Information messages

Information messages are used to inform the user that some wanted activity was approved or done successfully. Our application will provide these messages:

- **Wallpaper was successfully shared:** appears when the selected wallpaper was shared with other applications on social networks.
- **Wallpaper was set as account picture:** appears when the selected wallpaper was successfully set as the account picture.
- **Wallpaper was set as lock screen:** appears when the selected wallpaper was successfully set as the lock screen image.
- **Wallpaper was set as desktop wallpaper:** appears when selected wallpaper was successfully set as the desktop wallpaper.
- **Wallpaper was saved successfully. Check your Downloads folder:** appears when the selected wallpaper was successfully saved into the Downloads folder.

7.9.2 Error messages

Error messages are used to inform the user that a wanted activity was not done due to some kind of error. Our application will provide these messages:

- **You do not have a connection to the internet or your connection is broken. Please check your connection and try again later:** appears when the application does not have a connection to the internet. The internet connection is checked at the start of the application and the user initiates some action scenario for the selected wallpaper.
- **Service is temporary unavailable, please try again later:** appears when some unpredictable error happens or when there is a problem with the connection to the wallpaper's server.

8 PRODUCT'S DESCRIPTION IN UML

Application requirements and functionalities will be described by using the Unified Modeling Language as described in the theoretical portion of this thesis. UML provides us the possibility to describe functionalities in a quick and clear way, which can be interpreted by developers in companies all around the world.

8.1 Use case diagram

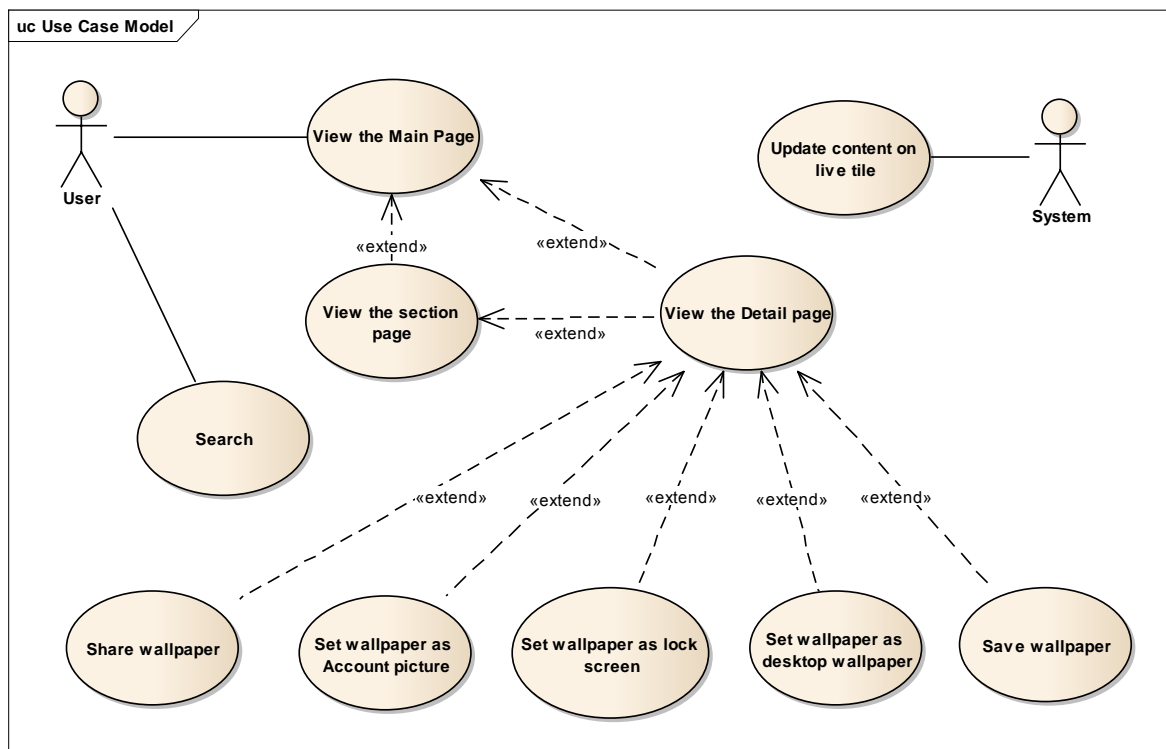


Figure42. Application's use case diagram

In the presented use case diagram two actors, the **system** and **user actors** are displayed. The **System actor** is associated only with one use case, because this use case details the scenario which is done automatically by the system without any interaction with the user. The **User actor** is associated with the rest of the use cases because they are always initiated by user interaction. Our application does not include the registration process so there is no need to divide the user actor in registered or paid type of accounts. The user is described as everyone who has installed our application.

Use cases represent individual situations when the user is using our application and generally describes scenarios. A list of use cases represent all possible scenarios which can be initiated by the user while interacting with our application. There are some cases that

have relations with others as clearly and visibly shown in the diagram. **Use case extensions** are use cases that describe particular activities with wallpaper extensions for viewing the wallpaper on the detail page. For the application this means that the app bar with the action command controls is available only on the wallpaper detail page. No action with wallpaper can be logically done until users select the wallpaper which they want to save, share, set as an account picture, set as the lock screen or set as the desktop wallpaper.

8.2 Use case scenarios

While use cases illustrate situations for using the application, use case scenarios represent every use case list of specific steps, depicting exactly how the user interacts with the application elements.

8.2.1 Search scenario

The Search scenario shows how the search use case can be initiated by the user from every page, including the main, section, detail and search result pages, while the application is running.

Table5. Search use case scenario

Step	Action
1	User slides from right edge of the screen
2	System shows charms
3	User tap on Search charm
4	System shows Search panel
5	User enter search query
6	User confirms entered search query
7	System show Search result page within application main canvas
8	User tap on selected wallpaper preview
9	System closes Search panel and open wallpaper detail page for selected wallpaper preview
10	Scenario ends

8.2.2 View wallpaper detail page scenario

This scenario explains the view detail page use case. As described in the use case diagram, this use case is an extension of the other use cases.

Table6. Viewwallpaper detail page

Step	Action
1	System shows main page.
2	User taps on Main category section label.
3	System shows Section page for selected category.
4	User taps on wallpaper preview
5	System shows wallpaper detail page
6	Scenario ends
Alternative scenario	
2.1	User taps on wallpaper preview
2.2	System shows wallpaper detail page
2.3	Scenario ends

In table 6 we show all required steps and the complete scenario for viewing the wallpaper detail page with both the primary and alternate steps for the different entry points.

8.2.3 Set wallpaper as account picture scenario

Table7. Set wallpaper as account picture scenario

Step	Action
1	System shows wallpaper detail page
2	User slides from the bottom edge of the screen
3	System shows app bar
4	User taps on Set as account picture icon
5	System set selected wallpaper as account picture

6	System displays proper information message
7	User click on close button
8	Scenario ends
Alternative scenario	
5.1	System does not set wallpaper as account picture
5.2	System shows proper error message
5.3	User click on close button
5.4	Scenario ends

8.2.4 Set wallpaper as lock screen scenario

Table 8. Set wallpaper as lock screen scenario

Step	Action
1	System shows wallpaper detail page
2	User slides from the bottom edge of the screen
3	System shows app bar
4	User taps on Set as lock screen icon
5	System set selected wallpaper as lock screen
6	System displays proper information message
7	User click on close button
8	Scenario ends
Alternative scenario	
5.1	System does not set wallpaper as lock screen
5.2	System shows proper error message
5.3	User click on close button
5.4	Scenario ends

8.2.5 Set wallpaper as desktop wallpaper

Table9. Set wallpaper as desktop wallpaper scenario

Step	Action
1	System shows wallpaper detail page
2	User slides from the bottom edge of the screen
3	System shows app bar
4	User taps on Set as desktop wallpaper icon
5	System set selected wallpaper as desktop wallpaper
6	System displays proper information message
7	User click on close button
8	Scenario ends
Alternative scenario	
5.1	System does not wallpaper as desktop wallpaper
5.2	System shows proper error message
5.3	User click on close button
5.4	Scenario ends

8.2.6 Save wallpaper

Table10. Save wallpaper scenario

Step	Action
1	System shows wallpaper detail page
2	User slides from the bottom edge of the screen
3	System shows app bar
4	User taps on Save as file icon
5	System saves selected wallpaper as a file

6	System displays proper information message
7	User click on close button
8	Scenario ends
Alternative scenario	
5.1	System does not saves wallpaper as a file
5.2	System shows proper error message
5.3	User click on close button
5.4	Scenario ends

8.3 Activity diagram

The Activity diagram gives in detail precisely how activity of sharing selected wallpaper will be accomplished. In Figure 43 the action states above that are not used in the activity diagram are divided by boundaries into two areas representing who is responsible for the action, the user or the system. The initial node is taken the moment the application is launched and the final node is at the end of this activity diagram.

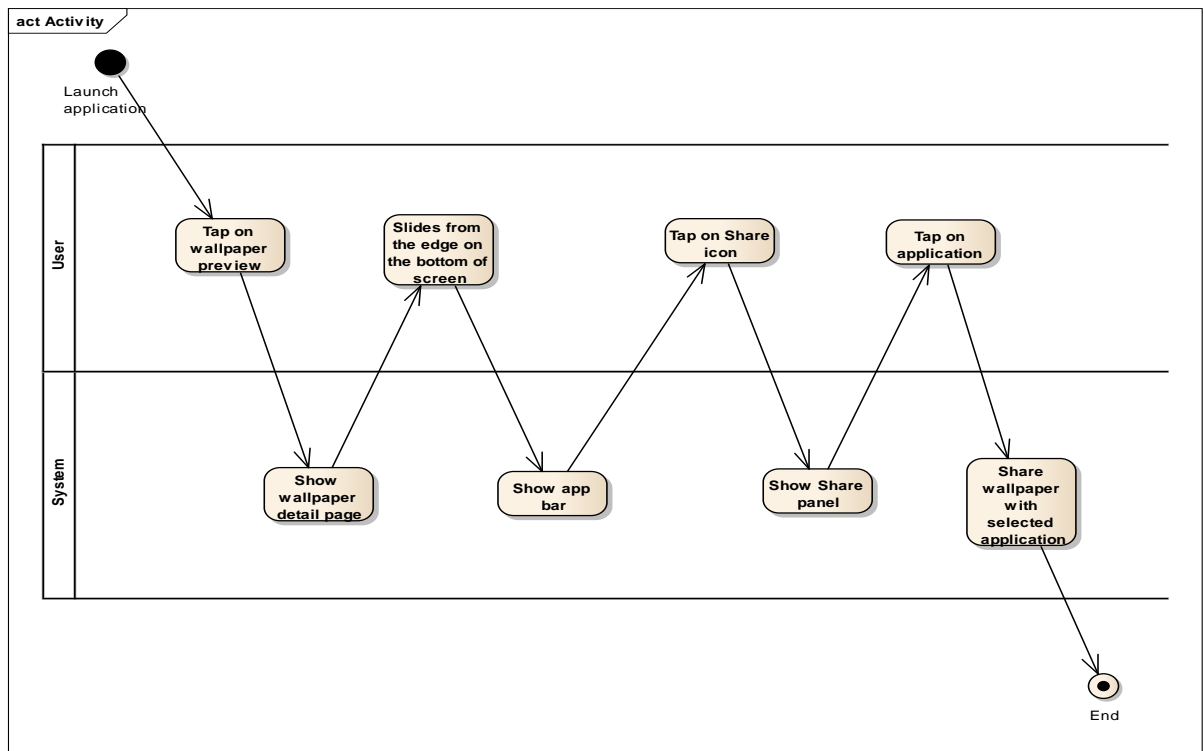


Figure43. Activity diagram for sharing wallpaper

8.4 Sequence diagram

Sequence diagrams help us clarify the order of interaction for selected scenarios. Figure 44 shows actions which have to be done to successfully achieve the goal of setting new desktop wallpaper. Some of these steps are invisible from user's point of view but they are all necessary for a successful finish of the scenario. There are 4 participants in this diagram the user, the application, the server and the OS desktop. The order of the individual actions is shown in the diagram and there are no other possibilities or exceptions.

1. Users select a command on the app bar displayed as Set as background wallpaper, and creates a request for our application.
2. The application asks the server to retrieve the selected wallpaper in high resolution. This must be done because in the application only the wallpaper preview is saved. By saving in the application, all wallpapers in high resolution will be absolutely inefficient when it comes to space consumption resulting in the server sending the wallpaper back to the application.
3. The application creates request for the operating system desktop to set new desktop wallpaper. When the operation is done the operating system confirms it back to the application.
4. Application shows the user the proper information message confirming that the new desktop wallpaper was set.

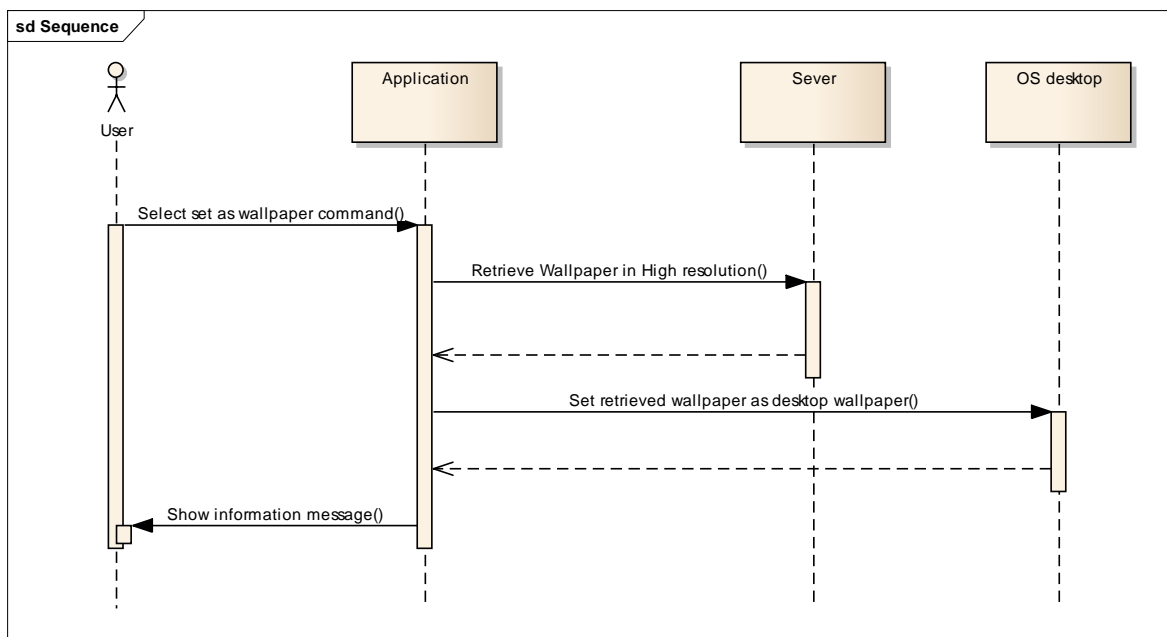


Figure44. Sequence diagram for setting new desktop wallpaper

9 PRODUCT'S RISK ANALYSIS

The following chapters explain the risks that can occur during the distribution of our application. Every chapter contains a short description of risks along with possible steps for minimizing or even eliminating these risks.

9.1 Users will not find our product

The first and one of the most important risks is that users will not find our application in the Windows Store. There can be several reasons for this risk to occur, but they all will result in the fact that our application will not be on the first page of results when the user will try to find our product or a similar application.

There are several ways we can improve the position of our application in the search results page, and they are all connected by ASO principles.

- **Keywords:** providing our description with the proper keywords is a never-ending process. We have to check the position of our application in the search results page periodically and also compare our description with competitors which are in a higher position. There are many tools for optimizing keywords and descriptions like MobileDevHQ or Searchman.
- **Updates:** no one knows exactly how applications are positioned on the search result page, but providing new updates for every application periodically will create a living organism. For end users it's important to know that the application that they will choose has history and is evolving continuously. With every update we are sending the users a message that we are working on fixing bugs and trying to improve the application according to the latest trends.
- **Ask for rating:** as we described in the theoretical section, ratings for every application represents users who are satisfied with our application. In the first phase, asking for a review is absolutely unpredictable because we do not know if there will be any problems. That is why asking for a review system should be implemented in later phases to ensure that there will be enough satisfied users which can give us a review.

9.2 There will be too many competitors

There is the possibility that in the future more and more applications with similar functionality can be released so this must also be taken as a risk. Currently similar applications are mostly developed by single developers or small companies with minimum resources for future development, but the possibility still exist that these are of the type of applications that will attract a big company with enough experience and sources.

The answer to this risk is simple, we must be better than our competitors. This basically means that in the future we must reserve some resources to continue with the development of this application. Presently no one knows what future trends will be because new ideas are created every day. What is important is how we will react to the new trend and thread in our application. Implementing new features as soon as they are requested, raising the satisfaction level of our users, and the rating of our product will be a crucial part in our future.

9.3 Bad rating from users

There are two possibilities for bad user ratings. The worst possibly being the user does not like our application at all. Risks that exist such as this have almost no pro-active steps for its elimination or at the very least the minimization of its influence. A better scenario would be that users are unsatisfied with some parts or only a specific function.

Reacting to these risks is time consuming. One step in solving these problems is by implementing new features and fixing bugs reported by users; fixing bugs seems to be a necessity for our application to survive. An important part of fixing bugs is the reactions from end users. Every time when a bug is fixed we have to keep the users informed, Not only in the application description but we also have to be more direct and communicate with them via email or social media. For an activity such as this there should be a dedicated person with expertise, probably a person from customer support or a marketing specialist for social media. Creating a page on Facebook and Twitter or any social network with live communication will have a positive effect on unsatisfied users, and with these networks we can ask them to remove their negative feedback or change it to a positive one.

9.4 Possible violation of trademarks or policies

Because the content of our application are wallpapers, there will always be the possibility of violating the policies or trademarks of other companies. Some wallpaper is free for distribution but others are protected by trademarks of companies or individuals.

One possibility to prevent this is to start producing our own wallpaper, but this can be very time consuming and graphic designers can be really expensive. A simple solution is to simply purchase wallpapers from the servers with paid images created by the designer. There are many services which focus on the paid distribution of images, but this solution is not free and will indeed cost us something. The price for wallpapers depends on their quality and resolution but in general it can range anywhere from between 1 and 50 dollars.

9.5 Change in windows store policies

There is always the possibility that policies and rules for applications in the Windows Store will be changed by Microsoft. This is an abstract risk but just because it's an abstraction does not mean it cannot still have minimum or even damaging consequences.

We currently do not know what can happen but we have to be prepared for virtually anything. This makes developing an application for a store interesting and exciting.

10 USED METHODOLOGY AND SPECIFICS

The last chapter is dedicated to summarizing the used methodology and techniques as well as comparing mobile applications and their specifics for common software development.

10.1 Specifics of mobile application

In comparison to standard software development disciplines like desktop applications and web presentations, the history of mobile applications is very young. Despite its youth, the development of mobile applications evolved into an important part of software development and has its own interesting specifics. Like all other aspects of mobile applications, specifics for mobile applications are dramatically changing.

10.1.1 Simplicity

The best known difference is its simplicity. Simplicity has its origins in the display size. This biggest limitation is obvious when we compare screen sizes:

- Common PC monitor: 22 inches.
- Laptop display: 13 inches.
- Mobile phone 4 inches.

For example, an iPhone 5 with its 1136 x 640 resolution size gives us at least some space but the need for simplicity is still high.

Simplicity principles have the biggest effect on UI. Standard elements like long menus, special effects on the mouse, and many overlay windows are completely useless when it comes to mobile applications. Every mobile platform has its own solution described in its guideline for developers. The basic principle of simplicity is to display only elements which are really necessary for users and to think twice about every used control; also every mobile device has its own hardware controls unified for all applications. Android and windows phones have an advantage by having 3 hardware controls for home, back and setting, while the apple mobile platform provides only a home hardware control.

10.1.2 Designing for different screens

Another obstacle is the need to design for different screens. This is possibly the main obstacle for developers because it generates many lines of codes and exceptions. When we are talking about mobile applications we must take into consideration the use of the

application on mobile phones and tablets. Designing patterns and UI can be completely different for the tablet because it has a bigger screen and surface for more elements. Many developers underestimate the difference between a tablet and a mobile device which can sometimes lead to negative ratings from end users. A common tablet with a screen that is 9.7 inches is two times bigger than the screen of a common mobile device, and an application that is designed only for mobile phones look empty on a tablet, almost as if it's a testing version or with some error.

10.1.3 Process of approving applications

Apple and Microsoft protect their stores by having an approving process. This means that every application released in the store must be approved before it is published for users. Google does not have an approving process and makes the android platform completely free, but every approach has its pros and cons.

The approving process guarantees users every application has been checked, is safe and meets all criteria. This has a big effect on user mentality because they have more trust in the whole platform and they know that every application has some kind of level. This process almost eliminates applications without any value and purpose. In addition, the approving process is very strict on user proposed elements and meeting guidelines; as a result all applications on the platform have similar usable elements which makes the whole platform clear and easy to use.

On the other hand, every developer must count on the approving process especially for the release schedule, because even if the application will be approved the first time, it is still time consuming. Any problems and remarks that occur in the approving process mean weeks of continuous development, and following the guidelines can also be a problem. When we are designing the mobile version of our desktop application we definitely have to use different UI, especially for the controlling elements. This can be confusing for users because they can be used to some behavior from different versions of the application.

10.1.4 Personal data of users

Every user must have an account to use the platform. Accessing the application through the store means the user's personal information is available and the developer must solve their problem. Using the application without registration, which is common behavior on the majority of web presentations, is not possible for mobile applications. Dealing with

personal information is set in the terms of use for every platform; every application has to explain which information is collected about the user and how the application handles the information. Developers can do it through individual terms of use and privacy policies for their application which users must agree before they begin using the app.

10.1.5 Payments

There is a completely different approach for the payment system. For desktop applications or web presentations, every developer has to provide its own payment system; this always leads to problems for using different systems followed by the lack of trust from the unknown.

Each mobile platform contains its own solution for its payment system. This is a significant difference because the end user provides their payment to the store and immediately gets what they ordered. In addition, the fact that mobile platforms are represented by some of the largest and world leading technological companies sets the level of trust at the highest level from the very beginning.

Every application clearly has its own description regardless of it being free or paid, so users can quickly decide if they want to pay or not. Apple and its app store has an interesting approach in dealing with payments; while registering for the platform the end user has to make an initial payment, so the information of not only the developers but all users is being checked and compared to their bank accounts.

10.2 Used methodology and elements

10.2.1 Benefits of used methods

Using a combination of wireframes with the Unified Modeling Language allows us to create and complete the exact specifications of the mobile application. Thanks to UML we can create a precise description of requirements, functionalities and scenarios for every possible aspect for the application.

Use case diagrams help us define general possibilities the user will incur while using the application, and use case scenarios allow us to define step by step exactly how the user will use the application.

When it comes to the specifications of application behavior, activity diagrams show its strength. Using boundaries, the activity diagram helps us to provide a clear description of the actions connected with the user and the system.

Sequence diagrams explain the amount of time consequences for actions are involved in setting new desktop wallpaper. A sequence diagram is not important from end user's point of view but it provides basic specifications for programmers responsible for coding wanted behavior for the application.

10.2.2 Analysis of used elements

Application's UI was designed according to specifics for mobile applications and guidelines for developing applications in the Windows store.

- **Simplicity:** to adjust an application to achieve maximum possible simplicity we use only elements that can be directly used on the displayed screen. On every screen only the element connected to the content is displayed. Any menus, settings and customizable parts to make using the application unified and direct are removed.
- **Content:** The hierarchical structure helps us to organize content into categories that is common behavior for all software products. The only content displayed is the wallpaper preview. This is based on the idea that from the user's point of view the most important information is if they do or do not like the wallpaper. Information such as resolution, rating, or size is redundant so they are not displayed. Only on the search results page is every wallpaper preview displayed, along with the wallpaper name so the user can match and confirm their results to their search query.
- **Navigation:** navigation is done completely on the canvas by a touching interaction. The user can navigate through pages by using only three elements, tapping on the wallpaper preview, main category section label, and back button. These elements are completely designed according to the Windows store's guidelines.
- **Command controls:** command controls are displayed only on the wallpaper detail page by using the app bar according to Windows store guidelines. This approach ensures that application controls will be simple and clear, without any redundant controls. One exception is the search control accessible via charms, which is also recommended in the guidelines for the Windows store application.

CONCLUSION

The subject of this diploma thesis was to create product specifications of an application designed for distribution in the Windows Store, and to show if there are any specifications for mobile applications compared to standard software like desktop application or web presentations.

In the first chapter of the Theoretical part the basic principles, which should always be reflected in software development and are based on ethical human principles, is explained. The other two chapters reflected in theoretical part are different. While UML is used for describing applications and is based on years of experiences, ASO is a young discipline dedicated specially for mobile applications and their distribution in modern application stores.

The Practical part reflects the situations from the entire theoretical part and brings them to life through product specifications for wallpaper application, beginning with project backgrounds and main ideas. Our application reflects the release of the new Windows 8 operating system and provides solutions for users who are having difficulty with getting used to the new UI, by combining both the desktop and mobile apps approach. The specification itself combines UML elements and wireframes. This combination allowed us to describe precisely every possible interaction from the user's point of view, including where every element is specifically placed by the designer. We were also able to specify the time consequences for a set of actions which resulted in helping the programmers complete certain specifications due to the use of sequence diagrams.

The principles of the simplicity application is designed with only 9 types of elements, represented by 7 icons, and 1 navigation element for the wallpaper category and wallpaper preview image.

While designing the mobile applications, we were able to find out some additional, interesting and different specifications for mobile applications. One of the most visible differences is the need to prepare a design for different size screens and types of view. Other interesting differences for mobile applications are the approving process, the new approach to personal data, and payments that are connected directly with distribution in the Windows store. These specifics, for first time, seem like obstacles but they play important roles in achieving success with an application in the real world.

To conclude, this thesis brings an interesting point of view due to the fact that the development of mobile applications definitely has their own specifications in comparison to desktop applications and web presentations, but it still maintains its software development roots. On one hand the development of mobile applications require an innovative approach and must reflect new trends, especially when it comes to UI. But on the other hand it shows that without hard work based on ethical principles and devotion to the end user's needs and wishes, mobile applications will not survive their first week.

CONCLUSION IN CZECH

Cílem této diplomové práce bylo vytvořit produktovou specifikaci aplikace navržené pro distribuci na Windows Store a popsat specifika mobilních aplikací v porovnání se standardními desktopovými aplikacemi nebo webovými prezentacemi.

V úvodu teoretické části popisujeme základní principy projektového vedení, které jsou založeny na základních morálních hodnotách. Tyto základní principy bychom měli cítit v jakékoliv fázi vývoje aplikací. Další dvě kapitoly teoretické části jsou navzájem odlišné. Jazyk UML, používaný pro popis funkcionalit aplikace, vychází z mnoha let zkušeností. ASO je mladá disciplína, která se zaměřuje na mobilní aplikace a zejména jejich distribuci skrze moderní aplikační tržiště.

V praktické části vycházíme z poznatků popsaných v předchozí teoretické části a úročíme je ve specifikaci moderní aplikace již od první kapitoly, která se zaměřuje na pozadí celého projektu a shrnuje hlavní myšlenky. Naše aplikace odráží spuštění nového operačního systému Windows 8 a nabízí řešení pro uživatele, kteří si těžko zvykají na razantní změny v ovládání a na nové prvky. Specifikace využívá diagramy a další prvky jazyka UML a kombinuje je vhodně s wireframy. Tato kombinace nám umožnila vytvořit přesný a detailní popis aplikace zahrnující všechny možné způsoby, jakými může uživatel aplikaci využít a kde přesně na ploše budou jednotlivé ovládací prvky umístěny. Diagramy jazyka UML nám umožnili dokonale popsat časovou posloupnost kroků, které musí samotná aplikace vykonat, aby splnila požadavek uživatele. Vytvořená specifikace tak poskytuje detailní popis nejenom z pohledu uživatele, ale také pro programátory a grafiky, kteří budou aplikaci vyvíjet.

Splnili jsme také základní požadavek na jednoduchost, který je na mobilní aplikace kladen. Aplikace je navržena za využití pouze devíti prvků tvořených 7 ikonami, tlačítkem zpět a náhledy jednotlivých obrázků jako navigačními prvky.

Při tvorbě návrhu aplikace jsme si uvědomili některá specifika mobilních aplikací, která je odlišují od desktopových aplikací nebo webových prezentací. Jedním z nejviditelnějších specifik je nutnost navrhovat mobilní aplikace pro různé druhy a velikosti obrazovek, se kterými se konečný uživatel může setkat a na kterých může být aplikace prezentována. Další zajímavosti jako schvalovací proces, přístup k uživatelským údajům nebo platební proces, jsou spojené přímo z distribucí moderních aplikací na tržištích. Tyto odlišnosti se

mohou zpočátku jevit jako velké překážky, ale hrají důležitou roli v rychlém rozšíření, čímž se právě moderní a mobilní aplikace vyznačují.

Na závěr nás přivádí tato práce k zajímavé myšlence, že sice vývoj mobilních aplikací má svá specifika, ale velice důležité je nezapomínat na základní principy, vycházející právě ze zkušeností desktopových aplikací a webových prezentací. Na jednu stranu vyžaduje vývoj mobilních aplikací inovativní přístup zvláště pro návrh uživatelských prostředí. Ale na druhou stranu nesmíme zapomínat na trpělivou práci a smysl pro detail, tolik typický pro desktopové aplikace, a tím dávat na první místo přání uživatelů. Jednoduše řečeno, pokud přestaneme dělat věci správně, tak nepřežije naše aplikace svůj první týden.

BIBLIOGRAPHY

- [1] Stellman, Andrew. Applied software project management. Sebastopol, CA : O'Reilly, c2006, xii, 308 s. ISBN 978-0-596-00948-9.
- [2] MILES, Russ a Kim HAMILTON. Learning UML 2.0. 1st ed. Beijing: O'Reilly, c2006, xiii, 269 s. ISBN 0596009828.
- [3] Pilone, Dan. UML 2.0 pocket reference. Beijing, Farnham : O'Reilly, c2006, vii, 128 s. ISBN 978-0-596-10208-1.
- [4] Jakob NIELSEN. Designing Web Usability: The Practice of Simplicity. Berkeley, CA: Peachpit Press, c2000, 419s.ISBN: 1-56205-810-X.
- [5] Windows Store [Online], c2013 [cit. 2013-05-01], Available from:
<http://www.windowsstore.com/>
- [6] Mobile app marketing tools by MobileDevHQ [Online], c2013 [cit. 2013-05-07], Available from: <http://www.mobiledevhq.com/>
- [7] Apptamin [Online], 2013 [cit. 2013-05-07], Available from:
<http://www.apptamin.com/>

LIST OF SYMBOLS AND ABBREVIATIONS

UML	Unified Modeling Language
OS	Operating system
AM	Ante Meridiem - before noon
PM	Post Meridiem - after noon
ASO	App Store Optimization
UX	User experience
UI	User interface
App	Application
SEO	Search engine optimization
ASO	Application store optimization

LIST OF FIGURES

Figure 1. Use Case	18
Figure 2. An actor	19
Figure 3. Human and machine actors	19
Figure 4. Use case generalization	20
Figure 5. Use case inclusion	20
Figure 6. Use case extension.....	21
Figure 7. Initial node.....	23
Figure 8. Final node	23
Figure 9. Action state.....	23
Figure 10. Activity diagram.....	24
Figure 11. Decision in an activity diagram.....	24
Figure 12. Partitions in activity diagram	25
Figure 13. Participants in sequence diagram	26
Figure 14. Event and message in sequence diagram	26
Figure 15. Activation bar in sequence diagram	27
Figure 16.Flat navigation system. Taken from: [5]	35
Figure 17.Hierarchical navigation system. Taken from: [5].....	35
Figure 18. Canvas. Taken from: [5].....	37
Figure 19. Charms. Taken from: [5]	37
Figure 20. App bar. Taken from: [5]	38
Figure 21. Press and hold interaction. Taken from: [5]	38
Figure 22. Tap interaction. Taken from: [5]	39
Figure 23. Slide interaction. Taken from: [5]	39
Figure 24.Swipe Interaction. Taken from: [5]	39
Figure 25.Pinch and stretch interaction. Taken from [5]	39
Figure 26. Splash screen wireframe.....	47
Figure 27. Square and wide static tile.....	48
Figure 28. Square and wide live tile	49
Figure 29. Main page full landscape view wireframe	50
Figure 30. Main page full portrait view wireframe	51
Figure 31. Main page in fill view	52
Figure 32. Main page in snapped view	52

Figure 33. Section page in full view	53
Figure 34. Section page in full portrait view	54
Figure 35. Section page in fill view	55
Figure 36. Section page in snapped view	55
Figure37. Detail page full landscape view wireframe	56
Figure38. Detail page portrait view	57
Figure39. App bar on Detail page.....	58
Figure40. Searching application with results.....	59
Figure41. Notification messages.....	60
Figure42. Application's use case diagram	62
Figure43. Activity diagram for sharing wallpaper.....	67
Figure44. Sequence diagram for setting new desktop wallpaper	68

LIST OF TABLES

Table 1. Scope of project content	15
Table 2. Use case documentation	22
Table 3. Splash screen sizes.....	48
Table 4. Application tile sizes.....	49
Table5. Search use case scenario	63
Table6. Viewwallpaper detail page	64
Table7. Set wallpaper as account picture scenario	64
Table8. Set wallpaper as lock screen scenario.....	65
Table9. Set wallpaper as desktop wallpaper scenario.....	66
Table10. Save wallpaper scenario	66