МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Інститут Прикладного системного аналізу

Кафедра Системного проектування

**Лабораторна робота № 2.**

З дисципліни

«Проектування інформаційних систем»

На тему

***«***Software Requirement Specifications***»***

Виконала

студентка групи ДА-61

Материнська Софія

КИЇВ

2019

1. **Мета роботи:** вивчити основні етапи створення передпроектної документації (SRS).
2. **Задача:** використати приклад SRS для створення передпроектної документації згідно індивідуальної теми для виконання лабораторних робіт.
3. **Результати роботи:**

Table of Contents

1. Introduction
   1. Purpose(Intended audience and reading suggestions)
   2. Scope
   3. Definitions, acronyms, and abbreviations
   4. References
   5. Overwiev
2. Overall description
   1. Product perspective
   2. Product functions
   3. User characteristics
   4. Constraints
   5. Assumptions and dependencies
3. Specific requirements

Appendixes

Index

1. **Introduction**

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

* 1. **Purpose**

*This subsection should*

*a) Delineate the purpose of the SRS;*

*b) Specify the intended audience for the SRS.*

The purpose of this document is to present a detailed description of the requirements for the “VisitIt” software. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

* 1. **Scope**

*This subsection should*

*a) Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);*

*b) Explain what the software product(s) will, and, if necessary, will not do;*

*c) Describe the application of the software being specified, including relevant benefits, objectives, and goals;*

*d) Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.*

The “VisitIt” is a mobile application which helps travelers to build a unique personalized tourist route by adding desirable places in a particular city they are visiting. The system will be designed to simplify traveling as users will be able to add, change and delete different places, cross out locations they have already visited, sort them in a sequence.

The application should be free to download from either a mobile phone application store or similar services.

* 1. **Definitions, acronyms, and abbreviations**

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the mobile phone application |
| Stakeholder | Any person who has interaction with the system who is not a developer. |
| DB | Database |
| DESC | Description |
| RAT | Rational |

* 1. **References**

*This subsection should*

*a) Provide a complete list of all documents referenced elsewhere in the SRS;*

*b) Identify each document by title, report number (if applicable), date, and publishing organization;*

*c) Specify the sources from which the references can be obtained.*

*This information may be provided by reference to an appendix or to another document.*

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998. <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>

* 1. **Overwiev**

*This subsection should*

*a) Describe what the rest of the SRS contains;*

*b) Explain how the SRS is organized.*

The remainder of this document includes two chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

1. **Overall description**

*This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.*

*This section usually consists of six subsections, as follows:*

*a) Product perspective;*

*b) Product functions;*

*c) User characteristics;*

*d) Constraints;*

*e) Assumptions and dependencies;*

*f) Apportioning of requirements.*

This section will give an overview of the whole system. The system will be explained in its context to introduce the basic functionality of it. It will also describe stakeholders that will use the system. At last, the constraints and assumptions for the system will be presented.

* 1. **Product perspective**

*This subsection of the SRS should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.*

*A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.*

*This subsection should also describe how the software operates inside various constraints.*

This project is independent and self-contained system for Android. It is intended to help tourists to create routs through unknown cities in a convenient way including their own preferences by adding places to visit. The software will be a grate application for those who loves total planning and has a limited amount of time so they can build the best way to discover new cities.

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used.

* 1. **Product functions**

*This subsection of the SRS should provide a summary of the major functions that the software will perform*

*Note that for the sake of clarity*

*a) The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.*

*b) Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.*

With the mobile application users will have an opportunity to create a list of locations they want to visit in a particular city and then place them in desirable sequence. It will be possible to add a place, edit it, delete and cross out(tick as visited).

* 1. **User characteristics**

*This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of the SRS.*

The program is designed for travelers. Every Android user will be able to use this application as no special skills are required.

* 1. **Constraints**

*This subsection of the SRS should provide a general description of any other items that will limit the developer’s options.*

The mobile application will be constrained by the capacity of the database.

The system should be in compliance with Accessibility and Security policy.

* 1. **Assumptions and dependencies**

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.

1. **Specific requirements**

*This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.*

* 1. **External interfaces**

*This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.*

* + 1. ***User interfaces***

Every user after opening an application is on the main page – see Figure 1. User can add a new country or choose one of previous.

After choosing a country it is possible to do the same sequence of choices with cities.

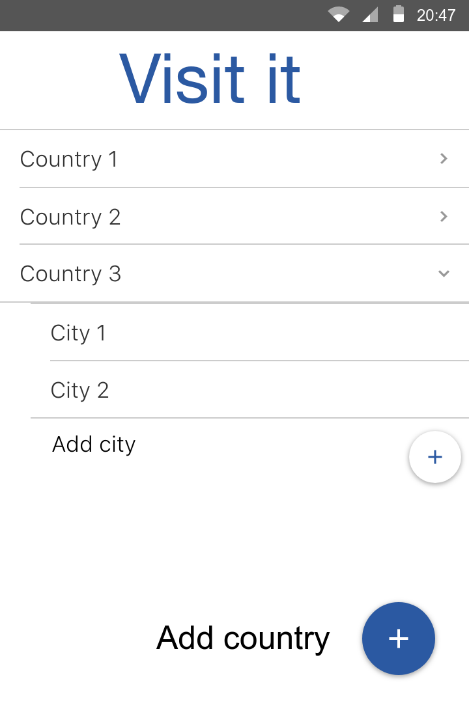
After a city is chosen you can add new place – see Figure 2 - or edit the one that already exists – see Figure 3.

Figure 1

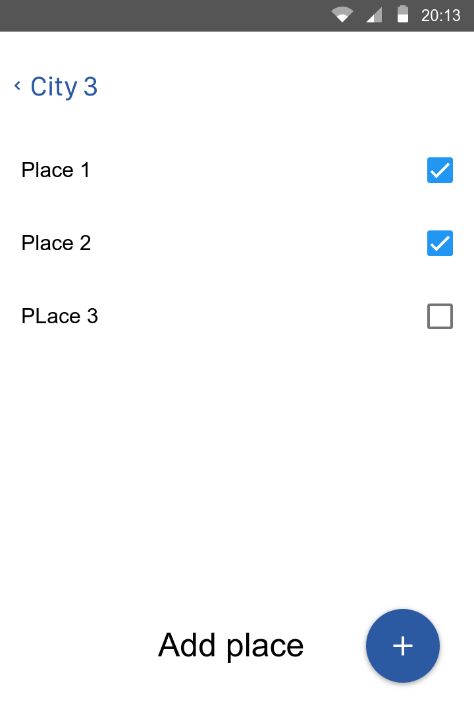
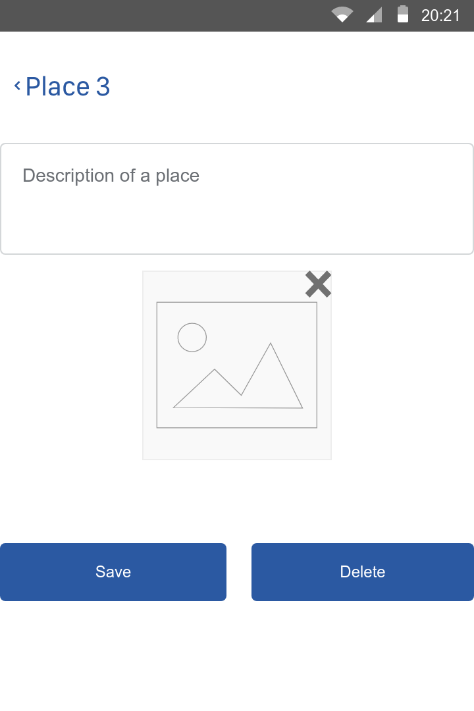


Figure 3

Figure 2

* + 1. ***Hardware interfaces***

It does not have any direct hardware interfaces***.***

* + 1. ***Software interfaces***

Operating system on the mobile device should be Android 5.1 or higher.

* + 1. ***Communication interfaces***

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems.

* 1. **Functional requirements**

*This section includes the requirements that specify all the fundamental actions of the software system.*

* + 1. ***Functional requirement 1***

ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.

RAT: In order for a user to download the mobile application.

* + 1. ***Functional requirement 2***

ID: FR2

TITLE: Add new country

DESC: A user should be able to add a new country that was not mentioned before. Needed input – country name. It should be checked if this country is already in the list.

RAT: In order for a user to add a new country.

* + 1. ***Functional requirement 3***

ID: FR3

TITLE: Add new city

DESC: A user should be able to add a new city that was not mentioned before in the border of particular country. Needed input – city name. It should be checked if this city is already in the list.

RAT: In order for a user to add a new city.

* + 1. ***Functional requirement 4***

ID: FR4

TITLE: Add new place

DESC: A user should be able to add a new place that was not mentioned before in the border of particular city. Needed input – place name, optional: description and photo. It should be checked if this place is already in the list.

RAT: In order for a user to add a new place.

* + 1. ***Functional requirement 5***

ID: FR5

TITLE: Edit chosen place

DESC: A user should be able to change information about current. Needed input – optional. Place can be deleted. Or changes can be saved.

RAT: In order for a user to edit mentioned places.

* 1. **Performance requirements**

*The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.*

N/A

* 1. **Design constraints**

*This section includes the design constraints on the software caused by the hardware.*

N/A

* 1. **Software system attributes**

*The requirements in this section specify the required reliability, availability, security and maintainability of the software system.*

* + 1. ***Reliability***

ID: QR1

TAG: SystemReliability

GIST: The reliability of the system.

SCALE: The reliability that the system update information correctly.

METER: Measurements obtained from 1000 cases of updating/creating new information about places.

MUST: More than 99% of cases.

PLAN: 100% of cases.

WISH: 100% of the cases.

* + 1. ***Availability***

ID: QR2

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability.

METER: Measurements obtained from 1000 hours of usage during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time.

WISH: 100% of the time.

* + 1. ***Security***

N/A

* + 1. ***Maintainability***

ID: QR3

TITLE: Application extendibility

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

RAT: In order for future functions to be implemented easily to the application.

ID: QR4

TITLE: Application testability

DESC: Test environments should be built for the application to allow testing of the applications different functions.

RAT: In order to test the application.

* + 1. ***Portability***

ID: QR5

TITLE: Application portability

DESC: The application should be portable with Android.

RAT: The adaptable platform for the application to run on.

* 1. **Other requirements**

**Appendixes**

**Index**

1. **Приймальні тести:**
2. Добавити нову країну. Добавити нове місто. Добавити нове місце. Зберегти.
3. В існуючу країну добавити нове місто. Добавити нове місце. Зберегти.
4. В існуючу країну і місто добавити нове місце. Зберегти.
5. В існуючу країну і місто добавити нове місце. Вийти не зберігаючи.
6. В існуючі країну і місто добавити нове місце, яке вже є в цьому місті цієї країни.
7. Редагувати наявне місце конкретної країни та міста. Зберегти зміни.
8. Вибрати наявне місце конкретної країни та міста. Видалити місце.
9. **Висновки: SRS** складається з трьох основних частин: вступу, загального опису та спеціальних вимог. Створення специфікації вимог до програмного забезпечення перед створенням програмного продукту покликане спростити процес розробки та прояснити мету створення та функції програмного забезпечення як для розробника так і для замовника ще до початку розробки, що дає також можливість витрачати менше ресурсів на внесення змін, натомість узгодити деталі до початку виконання, таким чином розподіливши розумно ресурси. Специфікація вимог передбачає написання максимально простого та зрозумілого опису для уникнення непорозумінь та можливості альтернативних інтерпретацій вимог будь-ким із учасників.