Software Requirements Specification

for

UrWallet Web-Application

Version 1.0

Prepared by

**Farrell Anandra Sadtyanto – 2201747622**

**Shonny Sholahudin – 2201823416**

**Aldandi Fitra Ramadhan – 2201830226**

**Gregorius Aldo Primantono – 2201846395**

Bina Nusantara University

2020

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of UrWallet Web-Applications is to make people to calculate income and expense easier.

## Document Conventions

We are using Times with font size 14 for the contents heading and Arial with font size 11 for the contents. For the header, we use Arial with font size 32. We use bold font for every contents heading.

## Intended Audience and Reading Suggestions

This Project help customers who have problems in managing finances to be able to choose how much to spend and their priorities. Also, This project is restricted within the university premises.

## Product Scope

People who have financial problems, especially people who are middle to lower income people with many needs and desires but still cannot save money even are in debt.

## References

* IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
* <http://web.itu.edu.tr/~tanriverdin/proje/SRSSample.doc>
* <http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc>

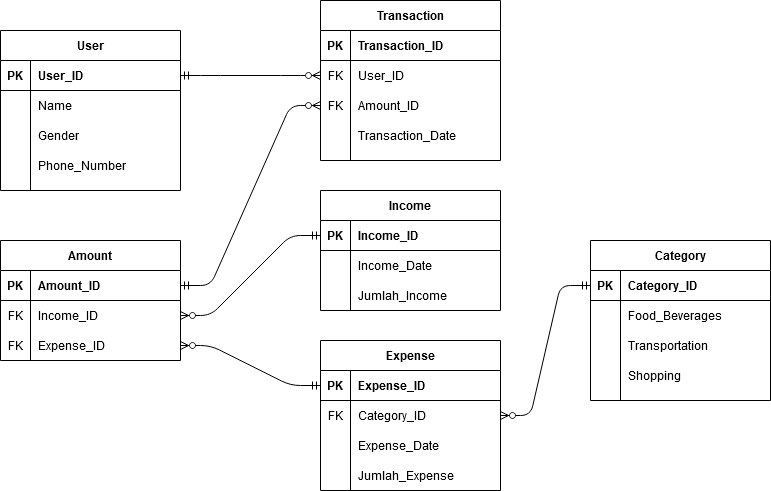
# Overall Description

## Product Perspective

Diagram

Description automatically generatedUrWallet System

## Product Functions

****Entity Relationship Model**

**2.2.1 User Use Case**

**A picture containing text, clipart

Description automatically generatedUse Case : Sign Up**

**Description :**

The user choose sign up on main menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Sever
2. The user choose sign up menu
3. The user fill the personal information such as name, gender, and phone number.

**A picture containing text, clipart

Description automatically generatedUse Case : Sign In**

**Description :**

The user choose sign in on main menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Sever
2. The user choose sign in menu
3. The user input phone number and password.
4. The system will sign in the user if the user input phone number and password correctly.
5. The system will not sign in the user if there any mistakes on phone number or password.

**A picture containing drawing

Description automatically generatedUse Case : Transaction**

**Description :**

The user choose transaction. There are incomes and expenses on transaction menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Server.
2. The user selects transaction button.

**A picture containing drawing

Description automatically generatedUse Case : Incomes**

**Description :**

The user choose incomes menu after transaction menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Server.
2. The user selects incomes button.
3. The user enters the income amount.

**A picture containing drawing

Description automatically generatedUse Case : Expenses**

**Description :**

The user choose expenses menu after transaction menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Server.
2. The user selects expenses button.
3. The user enters the expenses amount.

**Use Case : User Management**

Graphical user interface, application

Description automatically generated

**Description :**

The Admin choose user management. There are create account and delete account on user management menu.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The user connects to UrWallet Web Server.
2. The admin selects the user management button.

**2.2.2 Admin Use Case**

**Use Case : Delete account**



**Description :**

The Admin will delete user account if the user abuse UrWallet.

**Initial step-by-step description:**

For this use case, the user must be connected to the internet and connected to UrWallet Web Server.

1. The admin selects the delete account button.
2. The admin will email the user for abusing the UrWallet Web-Apps.

## User Classes and Characteristics

There will be two user classes, User and Admin.

User:

* User able to input incomes.
* User able to input expenses.
* User able to check transaction history.

Admin:

* Admin able to delete users account.
* Admin able to view users account.
* Admin able to view transaction of users.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Sign Up

|  |  |
| --- | --- |
| Use Case Name | Sign Up |
| XRef | - |
| Trigger | The user press sign up button on the main menu. |
| Precodition | 1. The user must be on the UrWallet main menu. 2. The user do not have UrWallet account. |
| Basic Paths | 1. The user inputs their name, gender, and phone number. 2. The user press sign up button. 3. The user will be logged in. |
| Alternative Paths | - |
| Postcondition | The user’s account is created. |
| Exception Paths | Sign up will fail if the user phone number is already registered. |
| Other | None |

## Sign In

|  |  |
| --- | --- |
| Use Case Name | Sign In |
| XRef | - |
| Trigger | The user press sign in button on the main menu. |
| Precodition | 1. The user must be on the UrWallet main menu. 2. The user must have an UrWallet account. |
| Basic Paths | 1. The user inputs their phone number and password 2. The user press sign in button. 3. The user will be logged in. |
| Alternative Paths | - |
| Postcondition | The user is logged in. |
| Exception Paths | Sign in will fail if the did not input phone number and password correctly |
| Other | None |

## Transaction

|  |  |
| --- | --- |
| Use Case Name | Transaction |
| XRef | - |
| Trigger | The user press transaction button on the transaction page. |
| Precodition | 1. The user must be logged in into UrWallet 2. The user must have an UrWallet account. |
| Basic Paths | 1. The user press transaction button 2. There will be 2 option, incomes and expenses. 3. The user press incomes or expenses button. 4. The user fill the amount of incomes or expenses. 5. If the user press the expenses button, there will be 3 categories of expenses, such as Food and Beverage, Transportation and Shopping. |
| Alternative Paths | - |
| Postcondition | The amount of incomes or expenses will be noted in UrWallet Web Apps |
| Exception Paths | The user can edit the amount of incomes or expenses if there a mistakes. |
| Other | None |

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>