|  |
| --- |
| BSHC4, BSHCE4, BSHBIS4, BSHBISE4 |
| Requirements Specification (RS) |
| Piggie.ie |

|  |
| --- |
| David Ward  10/1/2012 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 21/10/2012 | 1.0 | Create | David Ward |  |  |
|  |  |  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Stephan Weibelzahl | Lecturer |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc316977392)

[1.1 Purpose 4](#_Toc316977393)

[1.2 Project Scope 4](#_Toc316977394)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc316977395)

[2 User Requirements Definition 4](#_Toc316977396)

[3 Requirements Specification 4](#_Toc316977397)

[3.1 Functional requirements 4](#_Toc316977398)

[3.1.1 Use Case Diagram 5](#_Toc316977399)

[3.1.2 Requirement 1 <name of requirement in a few words> 5](#_Toc316977400)

[3.1.3 Requirement 2 <name of requirement in a few words> 6](#_Toc316977401)

[3.2 Non-Functional Requirements 7](#_Toc316977402)

[3.2.1 Performance/Response time requirement 8](#_Toc316977403)

[3.2.2 Availability requirement 8](#_Toc316977404)

[3.2.3 Recover requirement 8](#_Toc316977405)

[3.2.4 Robustness requirement 8](#_Toc316977406)

[3.2.5 Security requirement 8](#_Toc316977407)

[3.2.6 Reliability requirement 8](#_Toc316977408)

[3.2.7 Maintainability requirement 8](#_Toc316977409)

[3.2.8 Portability requirement 8](#_Toc316977410)

[3.2.9 Extendibility requirement 8](#_Toc316977411)

[3.2.10 Reusability requirement 8](#_Toc316977412)

[3.2.11 Resource utilization requirement 8](#_Toc316977413)

[4 GUI 8](#_Toc316977414)

[5 System Architecture 8](#_Toc316977415)

[6 System evolution 8](#_Toc316977416)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of services related to the Piggie.ie company. These services will include web services and mobile applications which allow people to securely allocate their funds into specific accounts which can facilitate withdrawals after a specific date.

The intended customers are individual consumer level customers who will be opening savings accounts to store small scale funds

## Project Scope

The scope of the project is to develop a web-service which securely allows a customer to amass funds in a distinct location and preventing their access to the location until a pre-defined date. The customer should be able to view the current balances of their funds and log in either through a website or mobile service. The system shall have a website front end, a mobile application on the Android platform and an API allowing it to be incorporated into third party websites.

This Requirements specification does not detail legal requirements which must be satisfied in establishing such a service.

## Definitions, Acronyms, and Abbreviations

Piggie.ie: The Piggie web service

Account: A membership account with piggie.ie

Savings account: an account set up to hold funds which are being saved through piggie.ie

Savings Pattern: how much and how often a user transfers money to a specific savings account.

Withdrawal Date: The date on which the user designates that their funds can be transferred for use.

……..

# User Requirements Definition

After conducting a brief market analysis it becomes apparent that currently a person’s options for savings are limited to three basic options. Savings through a bank account, savings through a credit union or savings in jars at home. Though discussions with peers it appears that there is a gap in the market for a service which allows specialization of savings tailored to specific tasks. This entails creating a service in which people can open accounts under specific names. They can designate how much is transferred into each account and the installment rate which it is transferred over on. They can designate the time at which this money then becomes available to them again and designate the account to which it is transferred back too. They can also check on balances periodically in a secure fashion through a website or mobile application.

# Requirements Specification

The user should be able create accounts and establish accounts without any specialised training. The service should be designed in a simple manner which follows a natural flow anyone familiar with setting up an account with any kind of service should be able to follow with ease while still maintaining confidence in the security of the service and believing it to be reputable in nature.

## Functional requirements

A user should be able to easily create an account.

A user should be able to log in to the service with their account.

A user should be able to create a savings account.

A user should be able to log in to see the current balances of their accounts.

### Use Case Diagram

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements.

### Requirement 1 <Create an account>

#### Description & Priority

The user should be able to create a membership account for the webservice. This is of paramount importance as all further functionality is based on this requirement being met.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

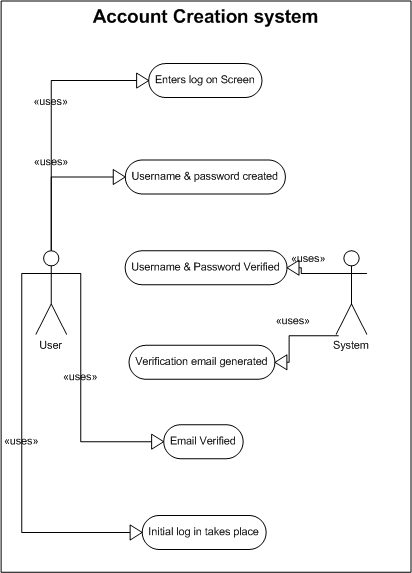
**Scope**

The scope of this use case is to define the process by which a user will create an account with piggie.ie

**Description**

This use case describes the account creation process for piggie.ie

**Use Case Diagram**



**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when the user logs on to the website www.piggie.ie…………

**Main flow**

1. The user should see a link which says create account.
2. The user should be redirected to a create login page
3. The user should be prompted to create a username and password
4. The system should verify that username and password are within acceptable criteria
5. The system should solicit an email from the user.
6. The system should forward the user an email for verification purposes.
7. The user will log in to their email account and use link to verify their account.
8. The system will terminate and exit.

**Alternate flow**

A1 : <title of A1>

1. The system will advise that the user cannot use this user name and password
2. The system will advise the user of the required criteria for username and password.
3. The user is prompted to pick a new username and password.
4. The use case continues on from point 4

**Exceptional flow**

E1 : <title of E1>

1. The system advises that the user has already used this email to create an account.
2. The system redirects user to the forgotten password screen and terminates.

**Termination**

The system now stores the user information for future log in’s and exits.

**Post condition**

The system goes into a wait state

### Requirement 2 <log in to service with account>

#### Description & Priority

When the user has created an account in order to use services they must log in to the service. This is vital to all use of the site

#### Use Case

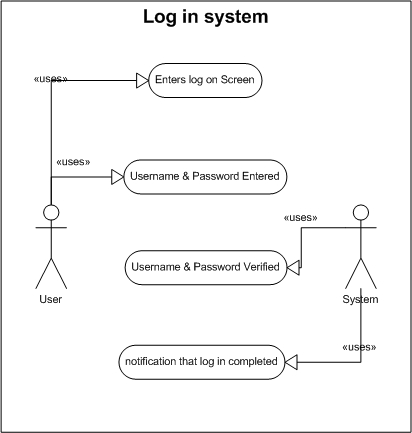
**Scope**

The scope of this use case is to describe the system by which a user logs into the system.

**Description**

This use case describes the login process for the piggie.ie website.

**Use Case Diagram**



**Flow Description**

**Precondition**

The user has created an account with the pigge.ie service.

**Activation**

This use case starts when the user goes to the piggie.ie website.

**Main flow**

1. The system presents a link on the site main page highlighting the log in screen.
2. The user clicks on the log in screen link.(See A1)
3. The system presents the user with a log in and password (See E1)
4. The user inputs their log in details created in as part of requirement one.
5. The systems verifies the users details and exits

**Alternate flow**

A1 : <the user enters incorrect details>

1. The user enters log in details
2. The system notifies that the user has entered incorrect details.
3. The system prompts the user with to enter new log in details
4. The use case continues at position 4 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The user enters invalid password and has forgotten the correct one.
2. The user is prompted to enter email used for verification.
3. The system exits and the user enters the password recovery system

**Termination**

Upon the user logging into piggie.ie the system exits.

**Post condition**

The user is logged in to piggie.ie

### Requirement 3 <Creation of a savings account>

#### Description & Priority

When the user has created an account and logged in to piggie.ie they must then create savings accounts.

#### Use Case

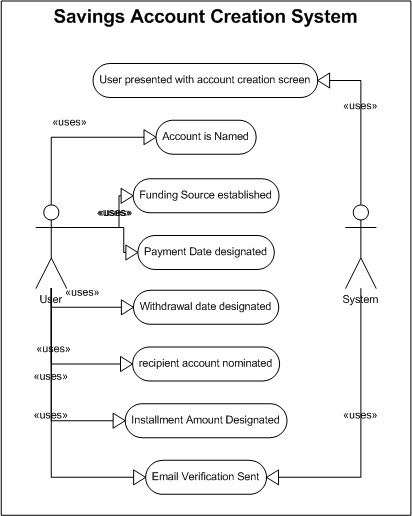
**Scope**

The scope of this use case is to define the process by which the user creates a savings account.

**Description**

This use case describes the process for the setting up of savings account

**Use Case Diagram**

**Flow Description**

**Precondition**

The user has created an account and used it to log in to the system.

**Activation**

This use case starts when an user has logged into the system.

**Main flow**

1. The user is presented with the account set up link.
2. The user is asked to name the account after what they are saving for
3. .
4. The user is asked for details from their paypal account.
5. The user is asked to define the date of the month in which they want funds to transfer into the account.
6. The user is asked to set a withdrawal date for which they can receive

their money back.

1. The user is asked to set the amount which will be transferred with each instalment.

**Alternate flow**

A1 : <title of A1>

1. The user enters incorrect information and wishes to go correct by pressing the back button.
2. The user is brought back to the proceeding position on the use case.

**Termination**

The system returns the user to the main screen where the user can now see the account with its status bar.

**Post condition**

The system goes into a wait state

### Requirement 4 <Log in Through Mobile Application>

#### Description & Priority

As lowest priority the user should be able to log into the site through an Android application.

#### Use Case

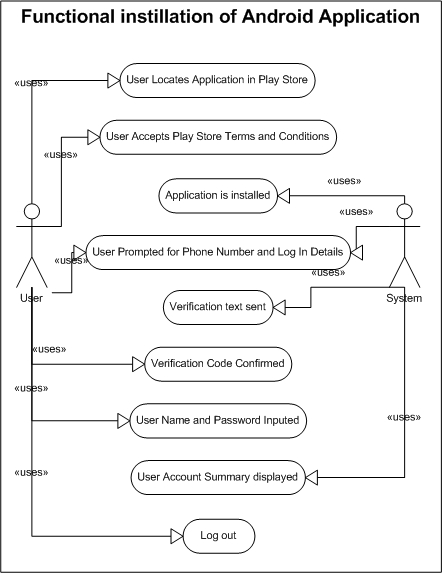
**Scope**

The Scope of this Use case is to set out the installation of Android Application

**Description**

This use case describes the functional installation of Android application for use by user.

**Use Case Diagram**

**Flow Description**

**Precondition**

The Use Case presumes that a user has already created an account with piggie.ie and established a savings account in their name.

**Activation**

This use case starts when an user wishes to install the Piggie.ie app on their Android phone.

**Main flow**

1. The user locates the piggie.ie application in the Play Market Place.
2. The user accepts terms and conditions of installation confirming order.
3. The system installs application to phone.
4. The user is prompted for their username, password and mobile phone number.
5. The system sends the user a verification text.
6. The user is asked for their verification code.
7. The user enters the code.
8. The user re-enters username and password.
9. The system displays user with screen showing names of accounts.
10. The user is shown summary details of account.
11. User logs out
12. Service exits and terminates.

**Alternate flow**

A1 : <verification not received.>

1. The system sends verification text message as per point 5 above
2. The user does not receive verification.
3. The user confirms that they have not received text message.
4. The system resends verification
5. The use case continues at position 6 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The user is not the owner of a Android phone
2. The user case terminates and exits at point 1

**Termination**

The system has closed and awaits use from point 8 in the main flow.

**Post condition**

The system goes into a wait state

**List further functional requirements here, using the same structure as for Requirements 1 & 2. Most systems would have at least five main functional requirements.**

## Non-Functional Requirements

Specifies any other particular non-functional attributes required by the system. Examples are provided below. **Remove the requirement headings that are not appropriate to your project.**

### Performance/Response time requirement

### Availability requirement

### Recover requirement

### Robustness requirement

### Security requirement

### Reliability requirement

### Maintainability requirement

### Portability requirement

### Extendibility requirement

### Reusability requirement

### Resource utilization requirement

# Interface requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include APIs, web services, shared memory, data streams, and so forth. Most systems would have a GUI. Add more subsections for other interfaces as reuired.

## GUI

Include mock-ups of the key pages or stages of the system. Explain how they are linked. Explain how you addressed above requirements in the design. It is important that the mock-ups are in line with the functional requirements above, e.g., if one of your requirements is “user registration” then one of the screens listed in this section should show a registration page.

## Application Programming Interfaces (API)

Explain which interfaces your system offers or which are used by your system. Examples include Google maps and Weka.

# System Architecture

Use a class diagram to outline the structure of the system. Explain briefly why you have chosen this architecture. You might want to use Visio or Rational Rose to create these.

# System Evolution

This section describes how the system could evolve over time.