Software Development Plan

SWE481-1803A-01

Instructor: Noura Abbas

Group 2:

Zach Sharpe

Lawrence Smith

Ryke Eccles

Alfred Gutierrez III

Table of Contents

[Project Outline 3](#_Toc521167423)

[Overview 3](#_Toc521167424)

[Development Methodology 3](#_Toc521167425)

[SCRUM Responsibilities 4](#_Toc521167426)

[Product Owner 4](#_Toc521167427)

[SCRUM Master 4](#_Toc521167428)

[Development Team 4](#_Toc521167429)

[Events and Deliverables 4](#_Toc521167430)

[Requirements 5](#_Toc521167431)

[Gathering process 5](#_Toc521167432)

[Profile Services 5](#_Toc521167433)

[Account Services 5](#_Toc521167434)

[Transfer Services 6](#_Toc521167435)

[Payment Services 6](#_Toc521167436)

[Design 7](#_Toc521167437)

[Architecture overview 7](#_Toc521167438)

[Components 7](#_Toc521167439)

[Login Page 7](#_Toc521167440)

[Main menu page 8](#_Toc521167441)

[Use Case Diagram 10](#_Toc521167442)

[Development 10](#_Toc521167443)

[Testing 11](#_Toc521167444)

[Test case Sample 1 12](#_Toc521167445)

[Test case Sample 2 13](#_Toc521167446)

[Test case Sample 3 14](#_Toc521167447)

[Test case Sample 4 15](#_Toc521167448)

[Project Schedule 15](#_Toc521167449)

[Risk Analysis 16](#_Toc521167450)

[Reference 17](#_Toc521167451)

# Project Outline

## Overview

The software objectives for the Bank management software, is to provide an easy to navigate user interface to interacting and operating common banking management operations. The software’s operations include profile management, bank account interaction, monitoring account transactions, generating account document statements, paying utility bills, credit card, and loan payments. The banking management software will be available to users by online websites and though a mobile application. The banking management systems primary operations will be managed and available for users to use through accounts in which they will need to create.

# Development Methodology

This project will utilize a common developmental strategy of Agile SCRUM. The development team will consist of a SCRUM Master, Product owner, and developer team. The proposed development will take 5 weeks and introduce complete, working, or implemented features at the end of each iteration. The SCRUM process will mitigate potential risks in sprints by incorporating bank customer and executive feedback. This method will allow for logging, and management updates to customer to help guide project process. The goal of this development strategy is to remain flexible and adapt to changes rather than stick to a strict strategy like waterfall.

Scrum will outline roles and responsibilities and help self-govern driving team principles:

* Team interaction and persons – Focus on communication and team interaction over developing solely with tools and software.
* Functionality Focus – Make sure the software is working and follow up with documentation when time allows.
* Customer Focus – Focus on the customer needs and problems as well as resources provided per iteration.
* Adapt to change – Be reasonable throughout the project and when requirements change, so should team development.

## SCRUM Responsibilities

### Product Owner

* Manage resources and priority of assigned tasking.
* Represent stakeholder’s wants and needs.
* Define requirements and acceptance.

### SCRUM Master

* Manages project pace and speed.
* Clearing objectives and obstacles that might prevent completion.
* Conduct, create and facilitate communication amongst developer team.

Development Team

Development Team consist of the following member’s Writers, Designers, Integration, Testers

* Write code for banking app
* Decide what and how work for each iteration should be completed.
* Backlog management and time conduciveness.
* Solutions, Risk Assessment, daily meetings.
* Design and functionality.

## Events and Deliverables

**Sprints –** Iterations consisting in weeks from 1 through 5 with team meetings daily and weekly to prioritize feature development, backlog tasking, and create solutions to issues. Sprints will be agile iterations accomplished with focus on implementation, testing and documentation. As well as review, evaluation and adaptation.

**Daily and Weekly Updates –** Scrum mater to coordinate and meet for 15-minute speed meets covering potential issues and feedback. During which we will review accomplishments, obstacles, and forward goals.

**Backlog** – Product owner will keep track of and keep priorities lined out with features, requirements, fixes and functionality as the main focus. This will also envelop the spring backlog for increments to focus on functionality of the next week.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Monitored Development * Constant feedback from all parties. * Adaptable team and project development. * Issues are discovered, logged, and resolved before causing conflict. * Deliverables are produced each iteration * High customer and team interaction * Minimal cost and time. | * Experienced and knowledgeable team members required. * Only works with small team to maintain speed and quality. * Acceptance and completion needs to be unanimous. * Thorough testing needed before integration. |

# 

# Requirements

Requirements gathering are the first major milestone of any software development lifecycle. Requirements are provided support for any development and it helps maintain a stable structure. Requirements are used to help determine what the software's intended purpose is. The following sections will help define the process when trying the gather requirements.

## Gathering process

The first step in gathering requirements is to identify the correct intended users of the software. By identifying the correct user they will provide a good insight into what is the need for the software from their perspective. Establishing interview questions can help the development team verify that they are good candidates for stakeholders your software development. After identifying who the stakeholders are going to be, establish a constant communication is important. Asking them what they are looking for in the software, by providing opening ended questions that establish an exact answer from them. This will provide an easy to understand respond from the interview regarding the requirements. The next step is to analyze the high-level business requirements and define the requirements in your own words with more detail and come up with visual representations. After coming up with detailed requirements report back to the stakeholders to verify that everyone is on the same level of understanding.

## Profile Services

•             As a user, I want to create an account so that I can access the banking managing operations.

•             As a user, I want to view my profile so that I can modify it at any time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Requirement** | **Requirement ID#** | **Type of testing** | **Requirement** |
| Profile Services | 1.1 | Functional | User can create an online profile |
|
| 1.2 | Functional | User can view profile information |
| 1.3 | Functional | User can login and log out of their existing profile |
| 1.4 | Functional | User can update and modify profile information |

## Account Services

•             As a user, I want to view my account activities so that I can see the different transactions that were done on my accounts.

•             As a user, I want the software to generate my account statements so that I can see a summary of my account actives

•             As a user, I want to be able to have the option to modify my visibility of my accounts and change account names so that it can be more convent for me to manage my accounts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Requirement** | **Requirement ID#** | **Type of testing** | **Requirement** |
| Account Services | 2.1 | Functional | User can view account activities |
|
| 2.2 | Functional | System can generate account statement for all unhide accounts. |
| 2.3 | Functional | User can manage visibility of accounts on web page |
|
| 2.4 | Functional | User can create nicknames for any accounts associated with profile |
|

## Transfer Services

•             As a user, I want to be able to transfer money from the software so that I can transfer money to internal or external accounts.

•             As a user, I want to be able to view transaction history so that I can better manage transactions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Requirement** | **Requirement ID#** | **Type of testing** | **Requirement** |
| Transfer Services | 3.1 | Functional | User transfer money to an internal and external account |

## Payment Services

•             As a user, I want to be able to add utilities bills to my payment options so that I can make payments though the software.

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Requirement** | **Requirement ID#** | **Type of testing** | **Requirement** |
| Payment Services | 4.1 | Functional | User can add common utilities bills to profile. |
| 4.2 | Functional | User can pay credit card payments. |
| 4.3 | Functional | User can pay loan payments. |

# Design

## Architecture overview

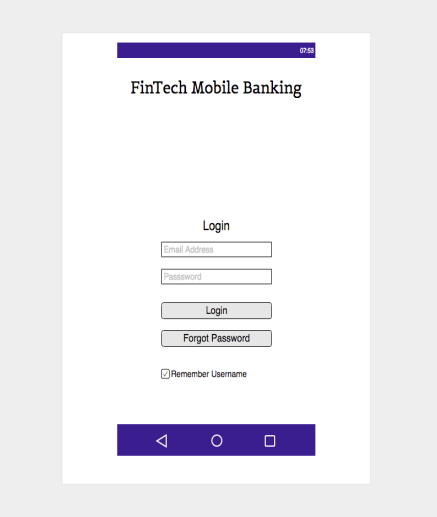
The customer is the user of the application. In order to access the Banking application they will go through the web portal of their mobile device. Once a customer accesses the application they can send or receive data from the customer’s primary bank database. The application can also work directly with an external portal for sending and receiving transactions to other banks. The external portal is then associated with the external bank portal to receive and send transactions to and from.

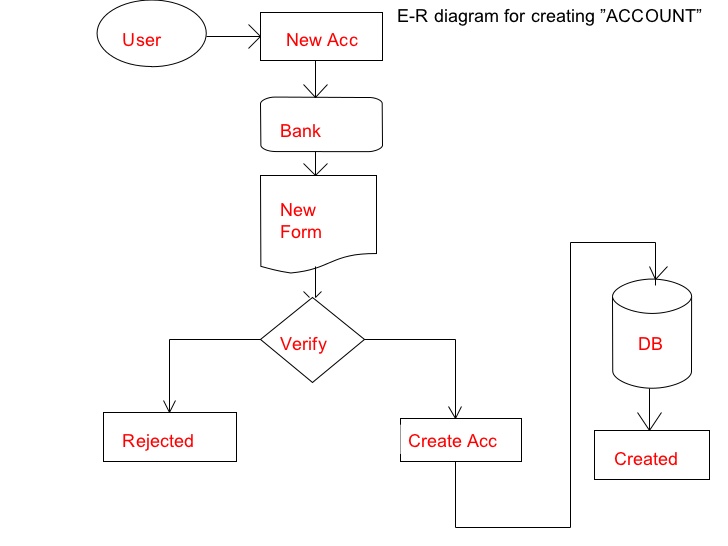
## Components

* Customer – The user of the application
* Banking Application – The banking application that will operate the bank management operations
* Banks Database – The data storage that will consist of customer profile, account and transactions data.
* External transaction portal – The portal that will be used to communicate the customer primary banks portal and the external bank portal
* External Banks portal – The portal that will is used by the external bank to process transactions too.

## Login Page

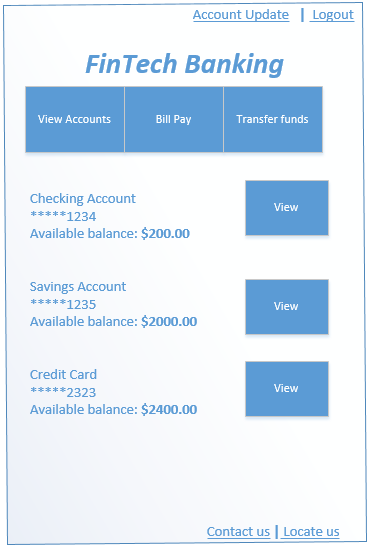
The following image shows the login page of the banking application. On this page the user can login or access the forgot password feature will process the user to enter an email that is assigned to a already created account and send an email with instructions to change the password.



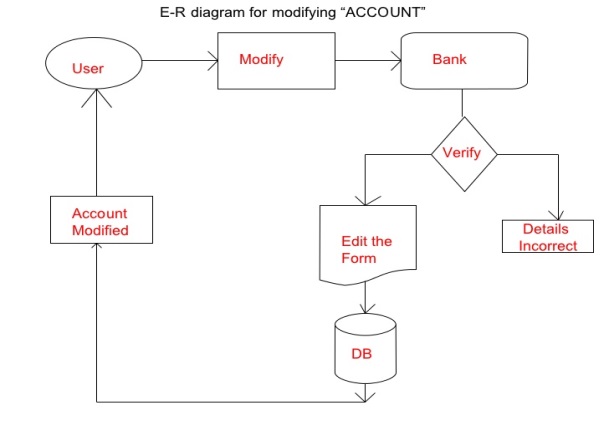
The following image is an ER diagram that shows the process of creating an account. 

## Main menu page

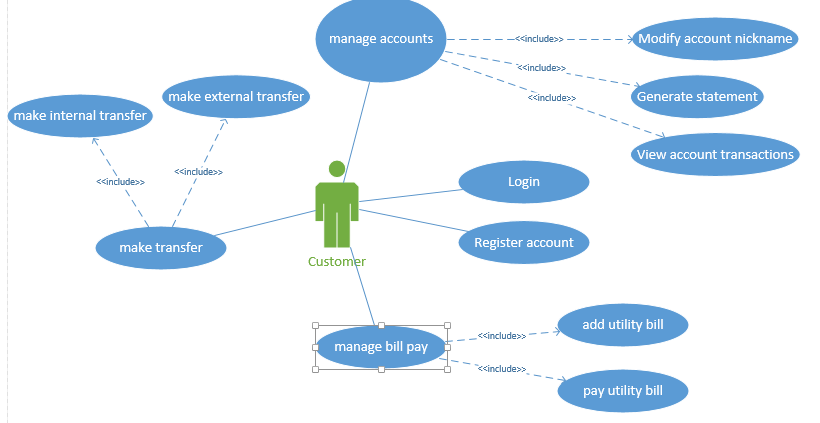
The following image shows the main page of the banking application after login. On this page the user can navigate throughout the application to operate the different banking management operations.



The following image is an ER diagram that shows the process that is used to make modification to an already existing account.



## Use Case Diagram



# Development

Development process will follow agile methodology. Working development will be separated into sprints. Each sprint is separated into its own phases. As mentioned before each sprint is consisted of three weeks to a month time period.

One of the key processes for development is tracking; the Kanban board will be used for this purpose. It will help to keep track of the working items progress during timeframe of each sprint and it will also help to manage what team member is working on what item. The Kanban board is spit into five sections, each section representing where the item is in development. The item will be managed by the scrum master, while providing a visual representation for the development. The following is example of how the Kanban will look. As an item is finished in one section then it will move to the next section changing who is responsible of the item.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Backlog** | **To Do** | **Planning** | **Build** | **Testing** | **Finished** |
|  | User story #123  Automatic timeout  Worked On by:  Peter Tomas | User story #124  Modify Profile  Worked On by:  Alan Frank |  | User story #125  Login to user account  Worked On by:  Sarah Jackson  User story #126  Failed Login to user account  Worked On by:  Janet Robinson |  |

Design is another process of development; it will take place prior to the building of the code. This section of development will go over the diagrams like flow charts, sequence, and use case diagrams to help developers understand the software or the item that will be added or modified during the sprint and how it affects/works in conjunctions with the other components.

Review is another process of development. Review will take place after testing and before deployment. During this time, a demo will be presented to the stakeholders. By providing a demo the stakeholder this will allow stakeholders to see a working prototype and verify first hand that the systems working item is working as intended. The review will follow by going over the testing status report. The testing status report will go over, testing and how it measured up to the requirements.

# Testing

Different types of testing will take place during different phase of development. During the Build phase, developers will be required to do unit testing to verify that code is working as intended from an internal prospective, verify that the logic in the code is working as expected. During testing phase, testers will work on system integration testing, to verify that the requirement is working as intended from a front end user prospective. After the initial sprint of the development is deployed new features will be added, while old features will be modified. Regression testing will be added to testing phase in future sprints to verify that the new changes don’t affect the already deployed functionality of the application.

|  |  |  |
| --- | --- | --- |
| Type of testing | Phase | Responsible for testing |
| Unit testing | During Build | Developer |
| System testing | During Test | Tester |
| Regression testing | After production launch  (During Test) | Tester |

## Test case Sample 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **1.1.1** | **Description**: The following test case is created to verify that a user can register an account with valid information and access the banking managing operations. | | |
| **Test Case** | **Register valid user** |
| **Step #** | **Step description** | **Expected Result** | **Actual Result** | **Status** |
| 1 | User navigates to the following URL “www.TestEnviBank.com/Login” | User is taken to the bank application login page | User is taken to the bank application login page | Pass |
| 2 | User clicks the register button | User is navigated to the create profile page | User was taken to the create profile page | Pass |
| 3 | User verity’s that the following fields are displayed with given labels.  First Name  Last Name  Street  Zip code  Country  State  City | Fields are displayed on create profile page | Fields are displayed on create profile page | Pass |
| 4 | User will enter the following information into each fields input box/ drop down menu.  First Name = Jane  Last Name = King  Street = 3215 Timber Hill  Zip code = 73412  Country = United State  State = Texas  City = San Antonio | User is able to enter or select the information into each of the fields | User was able to enter / select the given information into each field | Pass |
| 5 | User will click the continue button | User is taken to the account registration page. | User was taken to the account registration page. | Pass |
| 6 | User will verify that the following fields are displayed with given labels.  Social Security Number  Type of account  Account Number | Fields are displayed on account registration page | Fields are displayed on account registration page | Pass |
| 7 | User will enter the following information into each fields input box/ drop down menu.  Social Security Number = 65675753  Type of account = Checking  Account Number = 5675756754 | User was able to enter / select the given information into each field | User was able to enter / select the given information into each field | Pass |
|  | User will then enter the following information for the fields  Username: Tester1  Password: tester1Password | User is able to enter the following information for the given fields. | User was able to enter the following information for the given fields. | Pass |
| 8 | User will click the register button | User is taken to the congratulation page. | User was taken to the congratulation page. | Pass |
| 9 | User will verify that the following message is displayed  “User profile is created and is registered with accounts connected to given SSN.” | Message is displayed on the congratulation page. | Message is displayed on the congratulation page. | Pass |
| 10 | Verify that the first name is shown in the top right of the web page as  “Welcome: Jane | The label is displayed as expected with users first name shown | The label is displayed as expected with users first name shown | Pass |
| 11 | User clicks the Profile hyperlink located in the top right of the page. | User is taken to the update Profile page. | User was taken to the update Profile page. | Pass |
| 12 | User profile information is displayed correctly.  First Name = Jane  Last Name = King  Street = 3215 Timber Hill  Zip code = 73412  Country = United State  State = Texas  City = San Antonio | User profile fields is displayed and are prepopulated correctly as expected | User profile fields are displayed and are prepopulated correctly as expected | Pass |

## Test case Sample 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **3.1.1** | **Description**: The following test case is created to verify that a user can transfer a given amount of money from one account to another. | | |
| **Test Case** | **Transfer Money - Valid Amount** |
| **Step #** | **Step description** | **Expected Result** | **Actual Result** | **Status** |
| 1 | User navigates to the following URL “www.TestEnviBank.com/Login” | User is taken to the bank application login page | User is taken to the bank application login page | Pass |
| 2 | User will enter the following information into the given fields.  Username: Tester1  Password: tester1Password | User was able to enter the information into the fields. | User is able to enter the information into the fields. | Pass |
| 3 | User will click the login button | User was taken to the home page | User is taken to the home page | Pass |
| 4 | User will click the Transfer funds tab | User was taken to the transfer page | User is taken to the transfer page | Pass |
| 5 | Use will verify that the given fields with their labels displayed on the page.  From account  To account  Amount | The fields are displayed on the transfer page | The fields are displayed on the transfer page | Pass |
| 6 | Use will verify when clicking the input drop down menu button for the “From account” field the following options are displayed.  Checking \*\*\*1231: $ 100  Savings \*\*\*1211: $ 200 | The options are located in the drop down box. | The options are located in the drop down box. | Pass |
| 7 | User will select the Savings option for the “From account” field. | User was able to select the required option | User is able to select the required option | Pass |
| 8 | Use will verify when clicking the input drop down menu button for the “To account” field the following options are displayed.  Checking: $ 100 | The options are located in the drop down box. | The options are located in the drop down box. | Pass |
| 9 | User will select the Checking option for the To account field. | User was able to select the required option | User is able to select the required option | Pass |
| 10 | User will enter $50 for the “Amount” field. | User as able to enter the value for the field | User as able to enter the value for the field | Pass |
| 11 | Use will verify that the given fields show the following values.  From account: Savings: Savings \*\*\*1211: $ 200  To account: Checking \*\*\*1231: $ 100  Amount: $50 | The fields show the correct values | The fields show the correct values | Pass |
| 12 | User will click the transfer button | The following message will be displayed. “Transfer complete.” | The message is displayed | Pass |

## Test case Sample 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **4.2.1** | **Description**: The following test case is created to verify that a user can make a credit card payment though the app. | | |
| **Test Case** | **Make Bank Credit Card Payment – Valid Amount** |
| **Step #** | **Step description** | **Expected Result** | **Actual Result** | **Status** |
| 1 | User navigates to the following URL “www.TestEnviBank.com/Login” | User is taken to the bank application login page | User is taken to the bank application login page | Pass |
| 2 | User will enter the following information into the given fields.  Username: Tester2  Password: tester2Password | User was able to enter the information into the fields. | User is able to enter the information into the fields. | Pass |
| 3 | User will click the login button | User was taken to the home page | User is taken to the home page | Pass |
| 4 | User will click the Transfer funds tab | User was taken to the transfer page | User is taken to the transfer page | Pass |
| 5 | Use will verify that the given fields with their labels displayed on the page.  From account  To account  Amount | The fields are displayed on the transfer page | The fields are displayed on the transfer page | Pass |
| 6 | Use will verify when clicking the input drop down menu button for the “From account” field the following options are displayed.  Checking \*\*\*1111: $ 100  Savings \*\*\*2222: $ 200 | The options are located in the drop down box. | The options are located in the drop down box. | Pass |
| 7 | User will select the Savings option for the “From account” field. | User was able to select the required option | User is able to select the required option | Pass |
| 8 | Use will verify when clicking the input drop down menu button for the “To account” field the following options are displayed.  Checking \*\*\*1111: $ 100  Credit Card \*\*\*3333: $ 1000 | The options are located in the drop down box. | The options are located in the drop down box. | Pass |
| 9 | User will select the Credit Card option for the “To account” field. | User was able to select the required option | User is able to select the required option | Pass |
| 10 | User will enter $50 for the “Amount” field. | User as able to enter the value for the field | User as able to enter the value for the field | Pass |
| 11 | Use will verify that the given fields show the following values.  From account: Savings: Savings \*\*\*1111: $ 200  To account: Checking \*\*\*3333: $ 1000  Amount: $50 | The fields show the correct values | The fields show the correct values | Pass |
| 12 | User will click the transfer button | The following message will be displayed. “Payment Complete.” | The message is displayed | Pass |

## Test case Sample 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **2.3.1** | **Description**: The following test case is created to verify that a user can view bank accounts that are attacked to the user’s profile. | | |
| **Test Case** | **View Accounts** |
| **Step #** | **Step description** | **Expected Result** | **Actual Result** | **Status** |
| 1 | User navigates to the following URL “www.TestEnviBank.com/Login” | User is taken to the bank application login page | User is taken to the bank application login page | Pass |
| 2 | User will enter the following information into the given fields.  Username: Tester3  Password: tester3Password | User was able to enter the information into the fields. | User is able to enter the information into the fields. | Pass |
| 3 | User will click the login button | User was taken to the home page | User is taken to the home page | Pass |
| 4 | Verify that user can see the following tabs  View Accounts  Bill Pay  Transfer Funds | User was able to see that tabs on the current page. | User is able to see that tabs on the current page. | Pass |
| 5 | User can see the following accounts on the home page.  Checking: \*\*\*\*4321  Savings: \*\*\*\*2423  Credit Card: \*\*\*\*5343 | User was able to see the accounts on the home page. | User is able to see the accounts on the home page. | Pass |

# Project Schedule

. The following shows an outline of a typical sprint.

Requirements will be the starting point of the development process. This will consist of identifying the stakeholders, and gathering the team project team for the development. Once the team is formed, meeting will be taking place with the stakeholders to gather the requirements.

The creating of the product backlog will be the next point. The agile product backlog in scrum is a prioritized features list in which it contains a high level description of all the functionalities desired in the software product. Analyzing the requirements from a high level perspective and then prioritize the requirements. Then create user stories from the requirements and then add the user stories to the product backlog.

The next point in the development is the sprint process itself. Planning phase of the sprint will be to add the user stories into the current sprint. Design phase will go over the diagrams that will be used by developers to help understand the user story and how it. Build phase is where the developers will start to writing/ modifying the code adding the logic of the user stories. Testing phases is when testing of the application to verify that it is meeting the requirements of the user stories. The review phase it to show the stakeholders a prototype of the application so they can address questions to verify to see if its meets the requirements. The next phase is GO and NO/GO this is the approval meeting with the development and stakeholders to verify is what they saw during the demo and reviewing the testing documentation that they are stratified and approve of the deployment or not. The user stories will be what are required from the sprint. If the user stories are not complete during the initial sprint then it will move over to the next sprint with new user stories to follow. Each sprint follows the last to create a continuous integration development towards the software.

# Risk Analysis

TBD

# Reference

Agile Model & Methodology: Guide for Developers and Testers. (n.d.). Retrieved July 15, 2018, from <https://www.guru99.com/agile-scrum-extreme-testing.html>

Atlassian. (n.d.). What is scrum? - A brief introduction. Retrieved July 15, 2018, from <https://www.atlassian.com/agile/scrum>

Tutorials Point. (n.d.). UML - Component Diagrams. Retrieved July 24, 2018, from <https://www.tutorialspoint.com/uml/uml_component_diagram.htm>

Fakhroutdinov, K. (2014, January 04). UML Use Case Diagrams. Retrieved July 24, 2018, from <https://www.uml-diagrams.org/use-case-diagrams.html>

Cohn, M. (n.d.). Scrum Product Backlog and Agile Product Backlog Prioritization. Retrieved July 30, 2018, from <https://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog>

Daly, L. (n.d.). Kanplan: Where your backlog meets kanban. Retrieved August 4, 2018, from <https://www.atlassian.com/agile/kanban/kanplan>