**Insert Here Your Project Title**

Submitted by: InsertTeamNameHere

Submitted to: Hao Lac

ICET Department,

School of Engineering Technology and Applied Science

Progress Campus, Block A

Centennial College

Discipline: Software Engineering

Due Date: InsertDueDateHere

# **Declaration of Sole Authorship**

We, insertTeamNameHere, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed: insertMemberSignatureHere, insertStudentIDHere (programOfStudy)

Date: insertDueDateHere

# **Abstract**

Provide an abstract in this section in 200 words or less in paragraph format.

An accurate summary of the TR. State the main idea or thesis by answering questions such as:

● What is the TR about?

● Why is it significant?

● What should I do about it? [1]

**Table of Contents**

**Declaration of Sole Authorship 2**

**Abstract 3**

**List of Figures 6**

**List of Tables 7**

**1.0 INTRODUCTION 8**

**2.0 METHODOLOGY AND RESULTS 9**

2.1 Literature Review 9

2.2 Proposed Solution 9

2.3 User Role Modelling 10

2.3.1 Brainstorm and Group 10

2.3.2 Consolidated User Roles 11

2.3.3 Description of User Roles and Persona 12

2.3.4 Additional Documentation 13

2.4 Release 1.0 14

2.4.1 User Stories 14

2.4.2 Additional Documentation 18

2.4.3 Release Plan 1.0 19

2.4.4 Iteration Plan (Release 1.0) 20

2.4.5 Additional Documentation 21

2.4.6 Progress Monitoring 22

2.4.7 Acceptance Tests for Release 1.0 23

2.5.1 User Stories 25

2.5.2 Additional Documentation 26

2.5.3 Release Plan 2.0 27

2.5.4 Iteration Plan (Release 2.0) 28

2.5.5 Additional Documentation 29

2.5.6 Progress Monitoring 30

2.5.7 Acceptance Tests for Release 2.0 31

**3.0 CONCLUSIONS 32**

**4.0 RECOMMENDATIONS 33**

**CREDITS, LICENSE, AND REFERENCES 34**

Credits 34

License 34

References 34

**APPENDIX A (DESIGN DOCUMENT) 35**

**APPENDIX B (TEST PLAN) 36**

1.0 Introduction 36

1.0.1 Goals 36

1.0.2 Assumptions 36

1.0.3 Risks And Assets 36

Describe the elements that might positively influence testing on the project. 36

2.0 Scope 36

2.0.1 Features To Be Tested 36

2.0.2 Features Not To Be Tested 36

3.0 Testing Procedures 37

3.0.1 Test Objectives 37

3.0.2 Types Of Testing 37

3.0.2.1 Unit Testing 37

3.0.2.2 Integration Testing 37

3.0.2.3 Acceptance Testing 37

3.0.2.4 Stress Testing 38

3.0.2.5 Performance Testing 38

3.0.3 Testing Tools 38

4.0 Schedule and Deliverables 38

**APPENDIX C (END-USER & ADMINISTRATOR MANUALS) 40**

# **List of Figures**

Figure 1: Organizing the user role cards on a table [1]. 10

Figure 2: The consolidated role cards [1]. 11

Figure 3: Example of a “consolidated” low-fidelity prototype. Note that each “individual” low-fidelity prototype is developed for each user role [1]. 15

Figure 4: A story with notes providing additional detail [1]. 15

Figure 5: The revised front of a story card with only the story and questions to be discussed [1]. 16

Figure 6: Details that imply test cases are separated from the story itself. Here they are shown on the back of the story card [1]. 16

Figure 7: An example of a constraint story card [1]. 16

Figure 8: Possible electronic representation of a physical story card. 17

Figure 9: Iteration burndown chart for data from Table 5. 39

# **List of Tables**

Table 1: The Must-Have stories for Release x.y [1]. 17

Table 2: The Should-Have stories for Release x.y [1]. 17

Table 3: Disaggregated tasks per story [1]. 20

Table 4: Stories, acceptance tests, and contributors for Release 1.0 (Green=Accepted; Red=Rejected; Black=Not started). 23

Table 5: Progress and changes for all iterations. 39

# **1.0 INTRODUCTION**

Problem Statement:

In our increasingly digital world, book lovers and

bibliophiles are often faced with several challenges. The primary challenge is

finding a platform that allows them to discover, discuss, and share books based

on their personal preferences. Additionally, readers often struggle to find a

comprehensive platform that enables them to buy, rent, or exchange books

efficiently. Furthermore, the lack of a centralized forum to discuss and

analyze books with other enthusiasts often leads to a solitary reading experience.

Solution:

The solution is to develop an online book club website

that serves as a comprehensive platform for readers. This website will not only

allow users to discover and discuss books but also provide options to buy,

rent, and even suggest books based on user preferences. Additionally, users can

form or join groups to engage in meaningful discussions about their favourite

books.

System Capabilities:

· Register and store user information.

· Add and store users’ book preferences.

· Add and store information about books.

· Search and display books based on genre, author, price, etc.

· Allow users to buy, rent, or exchange books.

· Suggest books based on user preferences.

· Group formation and discussion forums.

· Direct message system for communication between users.

· Payment processing.

· Allow registered users to provide feedback and reviews.

Benefits:

· A comprehensive platform for book lovers, enhancing the

overall reading experience.

· Users save time and effort searching for books, as suggestions are based on their preferences.

· Users can easily buy, rent, or exchange books directly from the platform.

· Enables meaningful discussions and exchanges of ideas through book groups.

· Users can discover new books and authors based on recommendations from other users.

· Providing a platform for secure and transparent financial transactions.

Subsystems:

· Registration.

· Book cataloguing and search system.

· Payment system.

· Direct Message System.

· Book discussion forums.

The stakeholders and their roles:

· Readers: Discover, discuss, and share books; buy, rent,

or exchange books.

· Book owners: Sell, rent, or exchange books; add/modify/delete information about books.

· Group moderators: Lead book discussions, and manage group activities.

· Site admins: Manage site operations, user disputes, and overall maintenance.

# **2.0 METHODOLOGY AND RESULTS**

## **2.1 Literature Review**

There are numerous existing applications on the web that offer services related to book discovery and buying, such as Goodreads, Kindle, and Audible. These applications provide a platform where users can discover, read, or listen to books. However, these applications often fall short as comprehensive solutions for book lovers as they are mostly concentrated on selling books to users rather than fostering a community. They do not allow readers to exchange or rent books, nor do they offer a platform for group discussions about books.

## **2.2 Proposed Solution**

Our proposed online book club aims to serve as a comprehensive platform where users can discover, discuss, and share books based on their preferences. More importantly, through this platform, users will have the capability to buy, rent, or even exchange books with other users. The platform also offers the ability for users to form or join groups and engage in meaningful discussions about their favourite books. Furthermore, the platform will provide book suggestions based on user preferences and reading habits, thus enriching the overall user experience and promoting a more inclusive and engaging reading community.

A sample diagram can be found [here](https://drive.google.com/file/d/1aACSHAIBRRqlr9PMxfjaFi8UPg5G5P73/view?usp=sharing).

## 

## 

## 

## 

## **2.3 User Role Modelling**

### **2.3.1 Brainstorm and Group**

User Roles:

1.Readers

2. Book owners

3. Group moderaors

4. Site admins

Figure 1: Organization of User Roles

· Readers: Discover, discuss, and share books; buy, rent, or exchange books.

· Book owners: Sell, rent, or exchange books; add/modify/delete information about books.

· Group moderators: Lead book discussions and manage group activities.

· Site admins: Manage site operations, user disputes, and overall maintenance.

Benefits of the Online Book Club Platform:

1. Comprehensive platform for book lovers, enhancing the overall reading experience.

2. Time and effort-saving for users in searching for books, as suggestions are based on their preferences.

3. Easy buying, renting, or exchanging of books directly from the platform.

4. Enables meaningful discussions and exchanges of ideas through book groups.

5. Users can discover new books and authors based on recommendations from other users.

6. Providing a platform for secure and transparent financial transactions.

Subsystems:

1. Registration

2. Book cataloguing and search system

3. Payment system

4. Direct Message System

5. Book discussion forums

The proposed online book club website aims to address the challenges faced by book lovers by providing a comprehensive platform that offers various features and capabilities. Existing applications in the market lack certain functionalities, such as book exchange, group discussions, and recommendations based on user preferences. The proposed solution fills these gaps and aims to create an engaging and inclusive reading community.

Show the results of your brainstorming session for identifying initial user roles and how they are organized (see Figure 1). Discuss each user role identified and the arrangement of Figure 1.

Figure 1: Organizing the user role cards on a table [1].

### **2.3.2 Consolidated User Roles**

Show the consolidated user roles (see Figure 2). Discuss the results of Figure 2, focusing on why some roles were merged, removed, and/or added.

Figure 2: The consolidated role cards [1].

### 

### **2.3.3 Description of User Roles and Persona**

For each consolidated role from the above section 2.3.2, include detail that answer at least the following questions:

● The frequency with which the user will use the software.

● The user's level of expertise with the domain.

● The user's general level of proficiency with computers and software.

● The user's level of proficiency with the software being developed.

● The user's general goal for using the software. Some users are after convenience, others favour a rich experience, and so on.

Include personas here (optional).

### **2.3.4 Additional Documentation**

For this section, include the video(s) from your workshop showing how your team:

1. Brainstormed for the initial set of user roles.

2. Organized the initial set of roles.

3. Consolidated and condensed the roles.

4. Generated detailed description of each consolidated role.

Provide the file name and URL to the video(s) in your shared folder or YouTube channel.

## **2.4 Release 1.0**

### **2.4.1 User Stories**

The following are required for this section:

1. Show and discuss the results of your low-fidelity prototype generated during your story-writing workshop (a sample of a “consolidated” low-fidelity prototype is illustrated in Figure 3).

2. Provide your definition of story point.

3. Show the stories created during the story-writing workshop. You can submit scanned images of your index cards (both front and back). Figures 4 to 7 illustrates a single story with variation on the *Note*s (Figures 4 and 5), acceptance tests shown on the back of the index card (Figure 6), and a constraint, or non-functional requirement (Figure 7).

4. Prioritized stories based on the MoSCoW rule as illustrated in Tables 1 and 2 (see also *User Stories*deliverable).

Figure 3: Example of a “consolidated” low-fidelity prototype. Note that each “individual” low-fidelity prototype is developed for each user role [1].

Figure 4: A story with notes providing additional detail [1].

Figure 5: The revised front of a story card with only the story and questions to be discussed [1].

Figure 6: Details that imply test cases are separated from the story itself. Here they are shown on the back of the story card [1].

Figure 7: An example of a constraint story card [1].

Figure 8 illustrates a possible electronic representation of a physical story card. The left column represents the front of the card while the right column represents the back of the card.

|  |  |
| --- | --- |
| A Company can pay for a job posting with a credit card.    Note: Will we accept Discover cards?  Note for UI: Don’t have a field for card type (it can be derived from the first two digits on the card)    Estimate: 3 hrs. | Test with Visa, MasterCard and American Express.  Expected outcome: the system should automatically display a label of the card type.    Test with Diner’s Club.  Expected outcome: the system should prompt the user for a Visa, MasterCard or American Express card.    ...*<rest of the Tests follows>* |

Figure 8: Possible electronic representation of a physical story card.

Table 1: The Must-Have stories for Release x.y [1].

Table 2: The Should-Have stories for Release x.y [1].

### **2.4.2 Additional Documentation**

For this section, include the video(s) from your workshop showing how your team:

1. Brainstormed for stories and generated the low-fidelity prototype (story writing workshop).

2. Estimated stories using the Wideband Delphi approach.

3. Prioritized stories using the MoSCoW rule.

Provide the file name and URL to the video(s) in your shared folder or YouTube channel.

### **2.4.3 Release Plan 1.0**

The following are required for this section[1]:

1. Provide the product development roadmap.

2. Provide the iteration length and the release date.

3. The refine priorities of the Must- and Should-Have stories by organizing the stories into groups that have a high likelihood of being performed together.

4. The actual release plan.

5. Place the contents of your paper prototype in Appendix A (Design Document).

### **2.4.4 Iteration Plan (Release 1.0)**

The following are required for this section:

1. Present each iteration plan with tables showing disaggregated tasks per story; a sample is shown in Table 3. See also the *Planning an Iteration* deliverable.
2. Discuss any discrepancies between the estimated and actual ideal time required to complete the tasks for the Table mentioned above.

|  |
| --- |
| Table 3: Disaggregated tasks per story [1]. |

### **2.4.5 Additional Documentation**

For this section, include 1 of 4 videos from your Iteration Planning meetings (recall that you have a total of 4 Iteration Planning meetings)[2]:

1. Showing how your team disaggregated stories into their constituent tasks.
2. How developers on your team volunteer and take responsibilities for tasks.

Provide the file name and URL to the video(s) in your shared folder or YouTube channel.

### 

### **2.4.7 Acceptance Tests for Release 1.0**

The following are required for this section:

1. A table of stories and their associated acceptance tests for this Release as shown below in the sample in Table 5.
2. The link to your video demo for Release 1.0 stored either in a cloud drive, or your YouTube channel.

Table 4: Stories, acceptance tests, and contributors for Release 1.0 (Green=Passed; Red=Failed).

|  |  |  |
| --- | --- | --- |
| **Full description of user story** | **Acceptance test(s)** | **Name(s) of contributing Developer(s)** |
| As an User, I can … so that ….[3] | Test with inputs ….  Expected outcome: ...  Test with inputs ….  Expected outcome: ... | Susan Smith,  Jay Johnson |
| As an Administrator, I can … so that ….[4] | Test with inputs ….  Expected outcome: ...    Test with inputs ….  Expected outcome: ...  Test with inputs ….  Expected outcome: ... | Susan Smith,  Jay Johnson,  Shannon Shore,  George Gavinson |
| As an User, I can … so that …. | Test with inputs ….  Expected outcome: ...    Test with inputs ….  Expected outcome: ...  Test with inputs ….  Expected outcome: ... | Jay Johnson,  Shannon Shore,  George Gavinson |
| As an User, I can … so that ….[5] | Test with inputs ….  Expected outcome: ... | Shannon Shore |
| As a Guest, I can … so that …. | Test with inputs ….  Expected outcome: ...    Test with inputs ….  Expected outcome: ...    Test with inputs ….  Expected outcome: ... | Susan Smith,  Jay Johnson,  Shannon Shore,  George Gavinson,  Abbey Appleby,  Brian Bolt |

*<Insert url to video demo of Release 1.0 here>*

## **2.5 Release 2.0**

Release 2.0 has essentially the same structure as Release 1.0.

### **2.5.1 User Stories**

If your team wrote enough stories to cover up to or beyond Release 2.0 during your first story-writing workshop as described in the *User Stories* section 2.4.1, then your team will not need to hold a second formal workshop.

If a second workshop was held, submission for this section is the same as section 2.4.1.

### **2.5.2 Additional Documentation**

Include this section in your Technical Report only if your team required a second formal story-writing workshop. If a second workshop was held, submission for this section is the same as section 2.4.2.

### **2.5.3 Release Plan 2.0**

The requirements for this section are the same as section 2.4.3; update or add sections if required.

### **2.5.4 Iteration Plan (Release 2.0)**

The requirements for this section are the same as section 2.4.4.

### **2.5.5 Additional Documentation**

This section is required ONLY IF your team submitted materials for section 2.4.5.

### **2.5.7 Acceptance Tests for Release 2.0**

The requirements for this section follow the same requirements as in section 2.4.7 except acceptance testing is for stories allocated for Release 2.0 and incomplete stories subsequently moved from Release 1.0.

# **3.0 CONCLUSIONS**

A conclusion interprets the data found in the Body. It is reasoned judgment and not opinions. Consider the variables. Relate cause and effect. Analyze, evaluate, make comparisons and contrasts. Base the conclusion on fact.

# **4.0 RECOMMENDATIONS**

Recommendations are not required for all studies. They suggest a course of action and would generally be provided when there are additional areas for study, or if the reason for the TR was to determine the best action going forward.

# **CREDITS, LICENSE, AND REFERENCES**

## **Credits**

Provide any credits here. The following are examples:

Author of the template graphic layout : Hao Lac <haolac.at.centennial@gmail.com>

Author of the template explanation text : John Doe <john.doe@centennialcollege.ca>

## **License**

State the license granted with your system. For example:

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License".

## **References**

[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

# **APPENDIX A (DESIGN DOCUMENT)**

Traditional approaches to software development, in contrast to that of Agile approaches, place a great deal of emphasis on upfront design. The Agile approach to design is quick sessions that seek the simplest solution and then incrementally build on that solution. A quick design session can include the use of CRC cards that can ultimately lead to the generation of UML diagrams.

Using Agile approaches to software development does not mean you are limited to using only Agile techniques. If you feel that a technique (e.g., use case or interaction design scenario) is more suitable, or better conveys the features of your system to your users, then use it.

In this section, you are required to submit and discuss the following:

● A paper prototype of your application/system.

● Any design work your team has done in developing your system including CRC cards, UML diagrams, ERD diagrams, use cases, interaction design scenario, etc.

# **APPENDIX B (TEST PLAN)**

## **1.0 Introduction**

### **1.0.1 Goals**

Summarize the testing goals for the project.

### **1.0.2 Assumptions**

Any assumptions which may affect the understanding or execution of this plan should be recorded here.

### **1.0.3 Risks And Assets**

Describe the elements (software or hardware) that are not part of your application but still may impact its correctness and must be checked.

### Describe the elements that might positively influence testing on the project.

## **2.0 Scope**

### **2.0.1 Features To Be Tested**

Describe the features and functions that will be tested during the project. This should include functional and non-functional requirements.

### **2.0.2 Features Not To Be Tested**

Describe the features that will not be tested and reason why.

## **3.0 Testing Procedures**

Describe the testing procedures that the project will use. This includes the test lifecycle, types of testing, test objectives, and test criteria.

### **3.0.1 Test Objectives**

Describe the objectives of the testing process.

### **3.0.2 Types Of Testing**

Describe the types of testing that the project will use.

#### **3.0.2.1 Unit Testing**

Describe the strategy for unit testing of the individual subsystems. This includes an indication of the subsystems that will undergo unit tests or the criteria to be used to select subsystems for unit test. Test cases are NOT included here.

#### **3.0.2.2 Integration Testing**

Specify the integration testing strategy used. Describe the tests that will be performed in order to verify the interfaces between the subsystems of the software system. This section includes a discussion of the order of integration of subsystems. Test cases are NOT included here.

#### **3.0.2.3 Acceptance Testing**

Specify the strategy for testing the software once it has been deployed. This section includes a discussion of the order of acceptance by software function. Test cases are NOT included here.

#### **3.0.2.4 Stress Testing**

Identify the limits under which the program is expected to perform (memory constraints, disk space constraints, etc).

#### **3.0.2.5 Performance Testing**

Refer to the functional requirements that specify acceptable performance.

### **3.0.3 Testing Tools**

Describe the tools that you will use for testing.

## **4.0 Schedule and Deliverables**

Describe the test deliverables that will be created during the project lifecycle. Include two tables, one for the schedule of tasks, another for the list of deliverables:

● Acceptance test

● Unit test

● System/Integration test

● Stress test

● Performance test

● Screen prototypes

● Defect reports and summaries

● Test logs and reports

Describe the reports that will be generated by the testing process.

Examples include:

Test Summary Report - A final report of the testing results from the project. Can include items such as total number of test cases, number of test cases executed, % test cases passed, etc.

# **APPENDIX C (END-USER & ADMINISTRATOR MANUALS)**

In this section, include a user manual for your system/application. The user manual should include the following items:

1. Instructions on how to install and configure your system/application, documenting all external software dependencies that need to be setup manually.

2. A user guide for the administrator (use screen shots of your system/application and briefly discuss each screen shot).

3. A user guide for the normal user (use screen shots of your system/application and briefly discuss each screen shot).

# **APPENDIX D (PROGRESS MONITORING)**

Your team is required to report two items related to progress monitoring in this appendix. The first item is a table summarizing progress and changes during a release with supporting discussion; a sample is shown in Table 5. Notice in Table 5 that all iterations are shown per Release[6]. Also, see *Table 1* in the *Measuring and Monitoring Progress* deliverable.

|  |  |  |
| --- | --- | --- |
| Table 5: Progress and changes for all iterations [1].    The second item is an iteration burndown chart (see Figure 9) reflecting the data from Table 5.     |  | | --- | |  | | Figure 9: Iteration burndown chart for data from Table 5. | |

[1] See *The Release Plan* deliverable.

[2] Indicate which iteration the video corresponds to. If you decide to submit a video in Release 1.0, then you do not need to include an *Additional Documentation* section for Release 2.0.

[3] Green colour code indicates that all tests passed successfully as intended.

[4] Red colour code indicates that at least one test unintendedly failed.

[5] When all tests for a given story fails, this may suggest that implementation of the story has not even begun and indicates poor planning on the part of the team.

[6] For subsequent Releases, do NOT restart numbering the Iteration. For example, let us assume that we have another Release (i.e., Release 2.0), we would continue numbering our Iterations as *Iteration 5, Iteration 6,* and so on.

Formal report are always done in essay format. The bullet points are just instructional materials to ensure that the writeup meets the requirements from the bullet points.

Please remove all instructional materials as you progress through your technical report.