



Life Cycle Plan (LCP)

Newlette Coins

Team Number - 06

Members	Name	Email	Primary Role	Secondary Role
	Akshaya Ravichandran	ravichaa@usc.edu	Requirements Engineer	UML Modeler
	John Leibowitz	jleibowi@usc.edu	IIV&V	Quality Focal Point
	Nitin Surana	nsurana@usc.edu	Life Cycle Planner	Software Architect
	Remya Ramachandran	remyaram@usc.edu	Feasibility Analyst	Implementer
	Santhoshi Priyanka Gooty Agraharam	gootyagr@usc.edu	Project Manager	Tester

	Theerapat Chawannakul	tchawann@usc.edu	Builder	Implementer
	Vujjini Anuraag	vujjini@usc.edu	Implementer	Prototyper

Version History

Date	Author	Version	Changes made	Rationale
10/09/16	RR	1.0	Created initial draft of LCP from ICSM template	Initial draft for FCR ARB report
10/12/16	RR	1.1	Modified COINCOMO estimation.	Made revisions based on feedback from FCR ARB.
11/22/16	RR	1.2	Created section 6 of the LCP document	Completing LCP for the As Built package

Table of Contents

<u>Life Cycle Plan (LCP)</u>	3
<u>Version History</u>	4
<u>Table of Contents</u>	5
<u>Table of Tables</u>	6
<u>Table of Figures</u>	7
<u>1. Introduction</u>	7
<u>1.1. Purpose of the LCP</u>	7
<u>1.2. Status of the LCP</u>	7
<u>1.3. Assumptions</u>	7
<u>2. Milestones and Products</u>	8
<u>2.1. Overall Strategy</u>	8
<u>2.2. Project Deliverables</u>	9
<u>2.2.1. Exploration Phase</u>	9
<u>2.2.2. Valuation Phase</u>	9
<u>2.2.3. Foundations Phase</u>	10
<u>2.2.4. Development Phase</u>	10
<u>3. Responsibilities</u>	11
<u>3.1. Project-specific stakeholder's responsibilities</u>	11
<u>3.2. Responsibilities by Phase</u>	11
<u>3.3. Skills</u>	16
<u>4. Approach</u>	17
<u>4.1. Monitoring and Control</u>	17
<u>4.1.1. Closed Loop Feedback Control</u>	17
<u>4.1.2. Reviews</u>	17
<u>4.2. Methods, Tools and Facilities</u>	18
<u>5. Resources</u>	19
<u>6. Iteration Plan</u>	37
<u>6.1 Plan</u>	37
<u>6.1.1 Capabilities to be Implemented</u>	37
<u>6.1.2 Capabilities to be Tested</u>	38
<u>6.1.3 Capabilities not to be Tested</u>	39
<u>6.1.4 CCD Preparation Plans</u>	39
<u>6.2 Iteration Assessment</u>	41
<u>6.2.1 Capabilities Implemented, Tested, and Results</u>	41
<u>6.2.2 Core Capabilities Drive-Through Results</u>	42
<u>6.3 Adherence to Plan</u>	42

Table of Tables

<u>Table 1: Artifact Deliverables in Exploration Phase</u>	9
<u>Table 2: Artifact Deliverables in Valuation Phase</u>	9
<u>Table 3: Artifact Deliverables in Foundations Phase</u>	10
<u>Table 4: Artifact Deliverables in Development Phase</u>	10
<u>Table 5: Stakeholder's Responsibilities in each phase</u>	12
<u>Table 6: Development Team's Skills</u>	16
<u>Table 7: Tools used for the Project</u>	18
<u>Table 8: List of modules for Newlette Coins</u>	20
<u>Table 9: COCOMOII Scale Driver</u>	21
<u>Table 10: COCOMOII Cost Driver, Module 1: Login Page</u>	22
<u>Table 11: COCOMOII Cost Driver, Module 2: Register Page</u>	23
<u>Table 12: COCOMOII Cost Driver, Module 3: Game Page</u>	24
<u>Table 13: COCOMOII Cost Driver, Module 4: Loading Page</u>	26
<u>Table 14: COCOMOII Cost Driver, Module 5: Profile Page</u>	27
<u>Table 15: COCOMOII Cost Driver, Module 6: Leaderboard</u>	28
<u>Table 16: COCOMOII Cost Driver, Module 7: User Authentication</u>	29
<u>Table 17: COCOMOII Cost Driver, Module 8: Detonate</u>	31
<u>Table 18: COCOMOII Cost Driver, Module 9: Unit Tests</u>	32
<u>Table 19: COCOMOII Cost Driver, Module 10: Game Creation</u>	34
<u>Table 20: Construction iteration capabilities to be implemented</u>	37
<u>Table 21: Construction iteration capabilities to be tested</u>	38
<u>Table 22: Capabilities implemented, tested, and results</u>	41

Table of Figures

[Figure 1: COCOMO II Estimate](#)

36

1. Introduction

1.1. Purpose of the LCP

The LCP's main purpose is the following:

- Helps plan each iteration of the project's life cycle
- Helps monitor and manage available resources and skills throughout the duration of the project
- Tracks deliverables and risks encountered during the life cycle and facilitates risk evaluation, thereby enabling the development team to re-estimate costs and resources required, and modify the life cycle plan accordingly.
- Provides confidence to key stakeholders about the project and the development team, assuring them that all the risks and issues encountered are being dealt with.

1.2. Status of the LCP

The status of this LCP is currently at the version number 1.2, initial draft for Foundation Commitment Package. This version will be reviewed and updated based on feedback from other team members and changes our client suggests.

The latest version of this document will be delivered to the client.

1.3. Assumptions

- The duration of the project is 12 weeks in Fall 2016.
- All seven members of the development team will work on the project during these 12 weeks in Fall.

2. Milestones and Products

2.1. Overall Strategy

Newlette Coins is a new HTML5 game that has to be built from scratch and a major portion of its capabilities cannot be developed using existing Web Services or Non-Development items. Hence, the development team is following the Architected Agile Process of the Incremental Commitment Spiral Model.

Exploration phase

Duration: 09/07/16-09/16/16

Concept: Identify operational concept, system and software requirements, overall system architecture, and plan the software development life cycle.

Deliverables: Risk Defect Report, Progress Report, Project Plan, Client Interaction Report, and Win Conditions Report.

Milestone: Valuation Commitment Review

Strategy: One Incremental Commitment Cycle, Win-Win Negotiation Sessions, Risk assessment analysis.

Valuation phase

Duration: 09/17/16-09/26/16

Concept: Conduct win-win negotiation with client, identify and prioritize main objectives, define risks and constraints, analyze risks and explore mitigation strategies and alternatives, build an operation concept, work on software prototype and evidence of project feasibility, scrutinize and re-design project plan, define quality and configuration policy.

Deliverables: OOAD documents, initial architecture design, database ER diagram, Foundations Commitment Package initial draft

Milestone: Foundations Commitment Review

Strategy: One Incremental Commitment Cycle, Win-Win Negotiation Sessions, Risk assessment analysis, Planning Poker.

Foundations phase

Duration: 09/27/16 - 10/15/16

Concept: Analyze project status and plan, stabilize project progress to optimize deliverables' quality, develop prototype for high risk capabilities.

Deliverables: Working Prototype, UI/UX frontend design, Draft Foundation Commitment Package

Milestone: Development Commitment Review

Strategy: One Incremental Commitment Cycle, Risk assessment analysis.

Development phase**Duration:** 10/15/16- 12/05/16**Concept:** Project plan, first development cycle for developing the core requirements, second cycle mainly for testing and bug fixing, system transition.**Deliverables:** As Built Package, Final Package, Final working application.**Milestone:** Transition Readiness Review, Core Capability Drive, Development Commitment Review.**Strategy:** Two development cycles, Risk assessment analysis.

2.2. Project Deliverables

Mentioned below are the project deliverables during the different phases of the project:

2.2.1. Exploration Phase

Table 1: Artifact Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Team Website	09/14/2016	.html	Website Hosted Online
Client Interaction Report	09/16/2016	.pdf	Soft copy on team website
Win Conditions Report	09/26/2016	.pdf	Soft copy on team website
Jira tickets and report	Every Monday	Text	On google forms and Jira website
Project Plan	Biweekly on Wednesday	.mpp	Soft copy on team website
Progress Report	Biweekly on Wednesday	.xlsx	Soft copy on team website
Risk and Defect Report	Biweekly on Wednesday	.xlsx	Soft copy on team website

2.2.2. Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Top Risk Prototype Presentation	09/30/2016	.pdf	Soft copy on team website
Foundation Commitment Package ● Life Cycle Plan (LCP) ● Feasibility Evidence Description (FED)	10/17/2016	.pdf	Soft copy on team website

<ul style="list-style-type: none"> ●Prototype Report (PRO) ●Operation Concept Design (OCD) ●System and Software Architecture Description (SSAD) 			
Jira tickets and report	Every Monday	Text	On google forms and Jira website
Project Plan	Biweekly on Wednesday	.mpp	Soft copy on team website
Progress Report	Biweekly on Wednesday	.xlsx	Soft copy on team website
Risk and Defect Report	Biweekly on Wednesday	.xlsx	Soft copy on team website

2.2.3. Foundations Phase

Table 3: Artifact deliverable in Foundations Phase

Artifact	Due date	Format	Medium
Jira tickets and report	Every Monday	Text	On google forms and Jira website
Project Plan	Biweekly on Wednesday	.mpp	Soft copy on team website
Progress Report	Biweekly on Wednesday	.xlsx	Soft copy on team website
Risk and Defect Report	Biweekly on Wednesday	.xlsx	Soft copy on team website

2.2.4. Development Phase

Table 4: Artifact deliverable in Development Phase

Artifact	Due date	Format	Medium
As Built Package: <ul style="list-style-type: none"> ●Life Cycle Plan (LCP) ●Feasibility Evidence Description (FED) 	12/05/2016	.doc, .pdf	Soft copy on team website

<ul style="list-style-type: none"> ● Prototype Report (PRO) ● Operation Concept Design (OCD) ● System and Software Architecture Description (SSAD) ● Test Cases and Plan (TCP) ● Technical Manual (TM) 			
Jira tickets and report	Every Monday	Text	On google forms and Jira website
Project Plan	Biweekly on Wednesday	.mpp	Soft copy on team website
Progress Report	Biweekly on Wednesday	.xlsx	Soft copy on team website
Risk and Defect Report	Biweekly on Wednesday	.xlsx	Soft copy on team website

3. Responsibilities

3.1. Project-specific stakeholder's responsibilities

Our client is Crazy Cool Apps LCC, which is a startup company. They build and host HTML5 games that can be played on any browser on any device including laptops, desktops, mobile phones, and tablets. Our project is to build a new HTML5 game from scratch for the client according to their game play specifications. The client has to provide us with valuable feedback on the game design and assets during the development of the game. They will also need to maintain the game with the help of their developers and maintainers.

3.2. Responsibilities by Phase

Given below are the responsibilities of each member of the team during different phases of the project:

Table 5: Stakeholder's Responsibilities in each phase

Team Member / Role	Primary / Secondary Responsibility				
	Exploration	Valuation	Foundations	Development-Construction Iteration	Development - Transition Iteration
Name: Anuraag Vujjini Role: Implementer, Prototyper	Primary Responsibility Familiarize with Phaser.js. Implement simple user interaction capture on client side. Secondary Responsibility Identify top risks and design prototype for the same.	Primary Responsibility Work with the design team to incorporate the design assets in the frontend. Develop login and registration functionality for the game. Secondary Responsibility Identify key features that need to be demonstrated in the prototype. Create initial design of prototype.	Primary Responsibility Integrate frontend and backend. Test the user interaction, login and register functionalities. Secondary Responsibility Monitor prototype development.	Primary Responsibility Implement Edit Profile, leaderboard. Integrate UI assets. Fix bugs as and when they are logged by testers. Secondary Responsibility Identify key features that need to be demonstrated for CCD. Monitor development of game to ensure all key features are done before the CCD.	Primary Responsibility Ensure code is clean and well documented. Ensure application is seamlessly integrated with client's existing system and provide support as and when required. Secondary Responsibility Ensure application is seamlessly integrated with client's existing system and provide support as and when required.
Name: Akshaya Ravichandran Role: Requirements Engineer, UML Modeler	Primary Responsibility: Client Interaction. Analyze system architecture used in previous games built by client. Capture requirements of new game. Negotiate win-conditions with stakeholders. Secondary Responsibility Provide feasibility evidence for	Primary Responsibility Client Interaction. Make sure the architecture design meets specified requirements. Secondary Responsibility Work with system architect and develop UML diagrams. Help build design assets.	Primary Responsibility Client Interaction. Perform analysis of prototype to ensure adherence to stakeholder win conditions. Secondary Responsibility Ensure prototype adheres to UML diagrams designed earlier.	Primary Responsibility Client Interaction. Perform analysis of game features to ensure adherence to stakeholder win conditions. Secondary Responsibility Ensure game features adhere to UML diagrams designed earlier.	Primary Responsibility Client Interaction. Perform analysis of game features to ensure adherence to stakeholder win conditions. Secondary Responsibility Ensure game features adhere to UML diagrams designed earlier.

	gathered requirements.				
Name: John Leibowitz Role: IIV&V, Quality Focal Point	Primary Responsibility: Client interaction. Analyze current system of client's existing games. Secondary Responsibility: Analyze the requirements specified. Negotiate win-conditions with stakeholders.	Primary Responsibility Client Interaction. Analyze system architecture design. Secondary Responsibility Scrutinize the quality of the system's architecture, each development item, and design asset.	Primary Responsibility Client Interaction. Analyze prototype design. Secondary Responsibility Ensure quality of the prototype meets stakeholder requirements.	Primary Responsibility Client Interaction. Analyze system architecture design implemented. Secondary Responsibility Ensure quality of game being developed meets client's standards and meets all success critical stakeholders' requirements.	Primary Responsibility Client Interaction. Analyze developed system thoroughly and ensure smooth integration with client's existing system. Secondary Responsibility Ensure quality of game developed meets client's standards and meets all success critical stakeholders' requirements.
Name: Nitin Surana Role: Life Cycle Planner, Software Architect	Primary Responsibility: Analyze requirements and plan the project life cycle, and duration of each iteration. Secondary Responsibility: Evaluate system requirements. Design initial system architecture based on gathered requirements and existing client technology/platform to host game on.	Primary Responsibility Plan and estimate timeline for each iteration. Identify time and effort required for each task. Secondary Responsibility Develop detailed system architecture, and specify functionality of each module.	Primary Responsibility Develop detailed project iteration plan. Identify tasks to incorporate in each iteration. Identify responsibilities. Plan for backlog of work items. Secondary Responsibility Analyze progress of each module of the architecture and its ability to scale in size as the project progresses.	Primary Responsibility Develop detailed project iteration plan. Identify tasks to incorporate in each iteration. Identify responsibilities. Plan for backlog of work items. Secondary Responsibility Analyze progress of each module of the architecture and its ability to scale in size as the project progresses.	Primary Responsibility Develop detailed project iteration plan for the integration phase. Identify tasks to incorporate in each iteration. Identify responsibilities. Plan for a buffer period for maintenance of application as and when client requests, after which application will be handed over to client. Secondary Responsibility Ensure system architecture used for the application is

					well documented. Ensure application is seamlessly integrated with client's existing system and provide support as and when required.
Name: Santhoshi Priyanka Gooty Agraharam Role: Project Manager, Tester	Primary Responsibility: Analyze scope of project. Split tasks among team members on a weekly basis. Keep track of incomplete and accomplished tasks. Plan weekly team meetings. Keep check on project progress. Secondary Responsibility: Prioritize stakeholder requirements. Ensure that design/development meets major stakeholder requirements at all times.	Primary Responsibility Analyze scope of project. Split tasks among team members on a weekly basis. Keep track of incomplete and accomplished tasks. Plan weekly team meetings. Keep check on project progress. Secondary Responsibility Test the frontend and backend development thoroughly to ensure it fits the stakeholder requirements.	Primary Responsibility Analyze scope of project. Split tasks among team members on a weekly basis. Keep track of incomplete and accomplished tasks. Plan weekly team meetings. Keep check on project progress. Secondary Responsibility Test the prototype thoroughly and report bugs as and when discovered.	Primary Responsibility Analyze scope of project. Split tasks among team members on a weekly basis. Keep track of incomplete and accomplished tasks. Plan weekly team meetings. Keep check on project progress. Secondary Responsibility Continuously test the game thoroughly and report bugs in Jira as and when they are discovered.	Primary Responsibility Split tasks among team members on a weekly basis. Keep track of incomplete and accomplished tasks. Plan weekly team meetings. Keep check on project progress. Ensure integration with client's system runs smoothly and provide needed support. Ensure client gets required documents for maintenance of application. Secondary Responsibility Continuously test the game thoroughly and report bugs in Jira as and when they are discovered. Test for integration errors thoroughly.
Name: Remya Ramachandran Role:	Primary Responsibility: Perform risk and defect analysis	Primary Responsibility Analyze system architecture and	Primary Responsibility Conduct feasibility	Primary Responsibility Analyze team's progress and	Primary Responsibility Conduct feasibility

Feasibility Analyst, Implementer	based on requirements and resources available. Plan and explore mitigation strategies and other options. Prioritize risks identified. Secondary Responsibility: Familiarize with Java Spring. Design and build database and models for the game.	business workflow diagram and provide feasibility evidence for the same. Secondary Responsibility Implement controllers and functions to take care of business logic on the backend side of the game.	analysis on prototype created. Provide evidence to the client that the system will satisfy key stakeholder requirements based on the prototype analysis. Secondary Responsibility Integrate frontend and backend. Create tests to ensure all backend functionalities work as expected for prototype functionalities.	compare with amount of work left to be done. Note the requirements that are completed and weigh against those that are yet to be done taking into account the time left for the project's deadline and analyze the project's feasibility. Secondary Responsibility Implement Edit Profile, leaderboard. Integrate UI assets. Fix bugs as and when they are logged by testers.	analysis on final application developed. Provide evidence to the client that the system satisfies key stakeholder requirements based on the analysis. Ensure application is scalable and easily maintainable for client. Secondary Responsibility Ensure code is clean and well documented. Ensure application is seamlessly integrated with client's existing system and provide support as and when required.
Name: Theerapat Chawannakul Role: Builder, Implementer	Primary Responsibility: Analyze system requirements. Design development approach. Secondary Responsibility: Familiarize with Phaser.js. Implement simple user interaction capture on client side.	Primary Responsibility Scrutinize UI designs and oversee implementation process of the game. Secondary Responsibility Work with the design team to incorporate the design assets in the frontend. Implement REST API calls to the backend server to retrieve information	Primary Responsibility Oversee integration of frontend and backend. Assign testing of different aspects of the game to different members of the team after integration is complete. Secondary Responsibility Integrate frontend and backend. Test the user interaction and REST calls are	Primary Responsibility Oversee implementation of core functionalities. Assign testing of different functionalities to different team members. Secondary Responsibility Implement Edit Profile, leaderboard. Integrate UI assets. Fix bugs as and when they are logged by testers.	Primary Responsibility Oversee integration of the developed application with the client's existing system. Assign testing of different functionalities and bug fixing to different team members. Secondary Responsibility Ensure code is clean and well documented. Ensure application is seamlessly

		pertaining to the game.	sent properly to the backend.		integrated with client's existing system and provide support as and when required.
--	--	-------------------------	-------------------------------	--	--

3.3. Skills

The current and required skills for each team member for this project is listed below:

Table 6: Development Team's Skills

Team members	Role	Skills
Anuraag Vujjini	Implementer, Prototyper	<p>Current skills: C/C++, AJAX, jQuery, HTML5, CSS, SQL, XML, PHP, python.</p> <p>Required skills: Phaser.js, Java Spring, Photoshop, HTML5, CSS, SQL, AJAX, jQuery.</p>
Akshaya Ravichandran	Requirements Engineer, UML Modeler	<p>Current skills: Java, C/C++, HTML, CSS, SQL, Photoshop, Javascript, GME (Generic Modelling Environment), Axure</p> <p>Required skills: Photoshop, Java Spring, Phaser.js, Balsamiq/Axure, Microsoft Visual Paradigm.</p>
John Leibowitz	IIV & V, Quality Focal Point	<p>Current skills: Gimp (Open source Photoshop equivalent), Java, C/C++, VBA, Python, Javascript, HTML, CSS, SQL.</p> <p>Required skills: Gimp scripting/automation, IIV&V and Quality Focal Point role understanding.</p>
Nitin Surana	Life Cycle Planner, Software Architect	Current skills: java, javascript

		Required skills: game engine development, animation, game physics, image editing
Santhoshi Priyanka Gooty Agraharam	Project Manager, Tester	Current skills: Java, JavaScript, HTML5, CSS. Required skills: Leadership, Phaser.js, Java Spring, Photoshop, HTML5, CSS, SQL, AJAX, jQuery.
Remya Ramachandran	Feasibility Analyst, Implementer	Current Skills: C, C++, C#, Java, JavaScript, AJAX, jQuery, HTML5, CSS, SQL, XML, PHP, python. Required Skills: Phaser.js, Java Spring, Photoshop, HTML5, CSS, SQL, AJAX, jQuery.
Theerapat Chawannakul	Builder, Implementer	Current skills: Java, Javascript, PHP, Groovy, Angular.JS, jQuery, MySQL, Oracle, mongoDB, CSS Required skills: Javascript, Java, MySQL, Phaser.JS, CSS

4. Approach

4.1. Monitoring and Control

The project is monitored with a bi-weekly progress report and project plan. The progress report includes the top project risks, number of SLOC, COTS software and defects/concerns.

The following are used for monitoring and controlling the project's progress:

- Team meeting: The team meets after class every Monday and Wednesday to discuss tasks accomplished, tasks currently being worked on, and discuss potential roadblocks.
- Bi-weekly progress report: COTS, SLOC, priority risks, etc are tracked using this document.
- Bi-weekly project plan: Tasks are split and assigned to group members, and project progress is tracked using this.

- Bi-weekly technical debt reports: Used to keep track of all the debts that are yet to be taken care of.
- Weekly client meetings: Meet with the client once every week to discuss any requirement changes and to show the client the project progress.
- Jira: Documentation of work done and to be done by team members.
- Git: Progress in development is tracked using this.

4.1.1. Closed Loop Feedback Control

The following is used for feedback control within the team:

- WhatsApp Messenger: A WhatsApp group was created for communication among the team members, to schedule meetings, remind each other of upcoming deadlines, etc.
- Google Drive: A shared google drive folder was created to upload documents pertaining to the project for other team members to review and suggest modifications.
- Jira: Used to assign tasks to different team members and track tasks done.
- Peer Review Meetings: Every week the team meets to review progress and suggest modifications on various deliverables or design for the project, and help each other out in case of any major roadblocks.

4.1.2. Reviews

Types of reviews used by the team:

- Shared Folder on Google Drive: Every document prepared for the project is uploaded on the shared folder on google drive before finalizing it so that each member of the team can go through the document, review it and suggest any changes required.
- Peer review: Weekly team meetings are held to review each team members' work done for that week and each member gives his/her feedback and suggestions.
- Instructor/TA Reviews: Reviews from the instructor and TAs as and when various documents have been submitted have proved useful for the project's development.
- Client Meetings: Weekly meetings with the client helps get feedback from them about what is acceptable for the project and what should be reworked.

4.2. Methods, Tools and Facilities

Table 7: Tools used for the Project

Tools	Usage	Provider
Github	A version control system which acts as repository for the code checked in by different team members, allowing us to keep track of changes made.	Github
Bootstrap	Used to design better UI for the website	Open Source

Google drive	All documents required for the project are shared with team members through the team's drive so that everyone can download and review it.	Google
IntelliJ IDEA	Backend development of the project	JetBrains
Visual paradigm	Tool used to create UML diagrams	Visual paradigm
MySQL	Database	Open Source
Microsoft office	All documents for the project were created using MS Office; Word, PowerPoint, and Project in particular.	Microsoft
Brackets	Frontend development of the project	Brackets
WhatsApp Messenger	Communication within the team was facilitated through a Whatsapp group.	WhatsApp
HeidiSQL	Tool that enables fast and easy management of the database	HeidiSQL
Skype	Used to communicate with DEN student	Microsoft
Jira	To log time and effort spent on each task of the project by different members of the team	USC
Winbook	Used to identify and prioritize win conditions of all the success-critical stakeholders	USC
Project Website	All final documentation regarding the project is stored on the website.	USC
Balsamiq	Used to create project prototype design	Balsamiq

5. Resources

The calculation of effort and schedule estimation for our project using COINCOMOII is given in this section.

- Budget - \$0
- Programming language used – SQL, Java, HTML, CSS, AJAX, jQuery, php, JS.
- Project Duration – 12 weeks
- Number of developers - 7
- Number of modules – 10

The following table shows the main modules of development for the Newlette Coins application with the estimated SLOC.

Table 8: List of modules for Newlette Coins

No	Module Name	Brief Description	SLOC	REVL
1	Login Page	Gamers need to be able to login to their profiles to play the game. Their profile is used to keep track of their game history.	100	5%
2	Register Page	Gamers should be able to register for an account in the game to be able to play the game.	200	5%
3	Game Page	The main game page allows gamers to buy bombs, place them on the board and click detonate to win points.	900	10%
4	Loading Page	The loading screen should be displayed to the users while they wait for the game to load on their devices.	100	0%
5	Profile Page	This page should allow users to change/view their profile information.	130	5%
6	Leader Board	Game history of each user is saved. This can be used to find the 5 highest scorers and display the same to the users.	100	0%
7	User Authentication	This module will ensure each user logging in is properly authenticated before the game begins.	250	0%
8	Detonate	This module takes care of the pay table logic involved for calculating the number of points won by the user depending on the combination of	200	5%

		prize items he/she won when the bombs on the board were detonated.		
9	Unit Tests	This module takes care of testing the individual functionalities of the game.	150	0%
10	Game Creation	This module contains the logic to create a brand new game when the user starts the game.	100	5%

Table 9: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	Low	Most of the team members were new to Java Spring, only two had experience with it. None of the members had developed any HTML5 games using Phaser.js before.
FLEX	High	Client is open to suggestions and input from the team regarding the game UI/UX; business logic is fixed.
RESL	Nominal	Feasible risk elimination - prototyping high risk factors and buying more information from the client.
TEAM	Very High	Weekly team meetings ensure good communication and understanding of the project within the team. Peer review also helps solve roadblocks quickly. Good understanding between team members.
PMAT	Nominal	In the Capability Maturity Model - Defined level, with our own set procedure for development.

Table 10: COCOMOII Cost Driver, Module 1: Login Page

Cost Driver	Value	Rationale
RELY	Nominal	Moderate level of error or loss recovery if login page failed.
DATA	High	While testing, all invalid cases have to be handled so test data will be high.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	Nominal	Basic functionality, not very complex to code.
RUSE	High	Basic login code can be reused in other projects.
TIME	Nominal	Execution time isn't very high for login.
STOR	Nominal	User profile data doesn't take up a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Low	Team doesn't have previous experience developing authentication through a web application.
LTEX	Nominal	Team has had some experience with HTML5.
PLEX	Low	Most of the team members haven't worked with spring framework before.

TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 11: COCOMOII Cost Driver, Module 2: Register Page

Cost Driver	Value	Rationale
RELY	Nominal	Moderate level of error or loss recovery if register page failed.
DATA	High	While testing, all invalid cases have to be handled so test data will be high.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	Nominal	Basic functionality, not very complex to code.
RUSE	High	Basic registration code can be reused in other projects.
TIME	Nominal	Execution time isn't very high for registration.
STOR	Nominal	User profile data doesn't take up a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.

PCON	Very High	Team members will work on project till the end of the semester.
APEX	Low	Team doesn't have previous experience developing authentication through a web application.
LTEX	Nominal	Team has had some experience with HTML5.
PLEX	Low	Most of the team members haven't worked with spring framework before.
TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 12: COCOMOII Cost Driver, Module 3: Game Page

Cost Driver	Value	Rationale
RELY	High	High level of risk if game page fails in the middle of a game. Should ensure that none of the users loose their points or data when the game is back on. Has to be highly accurate.
DATA	Very High	Thorough testing has to be done for this module.
DOCU	High	Since this is the main module of the application, intensive documentation has to be done before handing the application over to the client.
CPLX	High	None of the team members had previous experience with Phaser.js or photoshop and this module uses both intensively to display various animations.
RUSE	Low	This module is specific only to this project and hence can not be used for other projects.

TIME	High	This is the main page of the game and it remains continuously displayed and modified as the user advances through the game.
STOR	Nominal	User's game data doesn't take up a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Very Low	None of the team members had previous experience with Phaser.js or photoshop and this module uses both intensively to display various animations.
LTEX	Very Low	Team has had no experience with phaser.js or photoshop before.
PLEX	Low	Team members haven't worked with phaser.js before.
TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 13: COCOMOII Cost Driver, Module 4: Loading Page

Cost Driver	Value	Rationale
RELY	Low	Low level of error or loss recovery if loading page failed.
DATA	Low	Not much data not required to test loading page.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	Low	Loading page is fairly simple compared to other modules.
RUSE	High	Module can be used for loading screen in other projects.
TIME	Nominal	Loading page will only be displayed before game starts, shouldn't take too long.
STOR	Nominal	Only design assets will be stored, doesn't require much memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	High	Loading page requires fairly simple logic to be applied.
LTEX	Nominal	Team has very little experience with phaser.js and photoshop, but proficient with HTML5.

PLEX	Nominal	Team has very little experience with phaser.js and photoshop, but proficient with HTML5.
TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 14: COCOMOII Cost Driver, Module 5: Profile Page

Cost Driver	Value	Rationale
RELY	Nominal	Moderate level of error or loss recovery if profile page fails.
DATA	High	While testing, all invalid cases have to be handled so test data will be high.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	Nominal	Basic functionality, not very complex to code.
RUSE	Low	Module is specific to this project's needs and can not be used in other projects.
TIME	Nominal	Profile page doesn't get displayed often.
STOR	Nominal	User profile data doesn't take up a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.

PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Nominal	Team doesn't have previous experience developing a profile page with avatars for every user.
LTEX	Nominal	Team has had some experience with HTML5.
PLEX	Nominal	Team has had some experience with HTML5.
TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 15: COCOMOII Cost Driver, Module 6: Leaderboard

Cost Driver	Value	Rationale
RELY	Nominal	Moderate level of error or loss recovery if leaderboard functionality fails.
DATA	Nominal	Testing for this module is fairly simple, doesn't involve much data.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	Nominal	Basic database querying, not very complex to develop.
RUSE	Low	Module is specific to this project's needs and can not be used in other projects.

TIME	Nominal	Leaderboard gets displayed only before loading a new game.
STOR	Nominal	User scores don't take up a lot of memory space.
PVOL	Nominal	Query doesn't get changed but the top scorers will vary as users keep playing.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Nominal	Team doesn't have previous experience developing a leaderboard functionality with a web application, but the functionality requires fairly simple logic.
LTEX	Nominal	Team has some experience with HTML5.
PLEX	Nominal	Team has some experience with HTML5.
TOOL	Very High	Tool used is brackets which is a time tested tool.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 16: COCOMOII Cost Driver, Module 7: User Authentication

Cost Driver	Value	Rationale
RELY	High	High level of risk if user authentication fails. Unregistered users shouldn't be allowed to play the game.

DATA	High	Module should be tested thoroughly for all types of end cases.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.
CPLX	High	None of the team members had previous experience with user authentication in Java spring framework.
RUSE	High	User authentication module can be reused for other projects done in the spring framework.
TIME	Nominal	Execution time isn't very high for user authentication.
STOR	Nominal	User details don't take up a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Low	Team doesn't have previous experience developing user authentication in Java spring.
LTEX	Nominal	Team has experience with Java.
PLEX	Low	Team doesn't have previous experience developing user authentication in Java spring.
TOOL	Very High	Tool used is IntelliJ Idea, MySQL and HiediSQL which is are well established tools used in industries for development.

SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 17: COCOMOII Cost Driver, Module 8: Detonate

Cost Driver	Value	Rationale
RELY	High	High level of risk if detonate fails. Calculation of prizes won by users based on the pay table is done in this module. Has to be highly accurate.
DATA	Very High	Module should be tested thoroughly for all types of end cases and invalid cases.
DOCU	High	This is one of the most important modules of the application and needs to be documented well before handing it over to the client.
CPLX	High	The calculation for the prizes won by user needs to be done accurately depending on the pay table combination.
RUSE	Low	This module is built specifically for this project and can not be used elsewhere.
TIME	Nominal	This module will be used only when the user clicks on the detonate button to see the prizes he/she has won.
STOR	Nominal	Doesn't require a lot of memory space to store the prizes won by the user for a game.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.

PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Low	Calculation for the pay table has to be accurate and the team hasn't had any previous experience building a game of this kind before.
LTEX	Nominal	Team has experience with Java.
PLEX	Low	Team doesn't have previous experience developing in Java spring.
TOOL	Very High	Tool used is IntelliJ Idea, MySQL and HiediSQL which are well established tools used in industries for development.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 18: COCOMOII Cost Driver, Module 9: Unit Tests

Cost Driver	Value	Rationale
RELY	Nominal	Moderate level of error or loss recovery if unit tests fails. Test failures help find other potential bugs in the system.
DATA	Very High	Module will be used to test each part of the system thoroughly using data for end cases, invalid cases, etc.
DOCU	Nominal	Right amount of documentation for maintenance purpose when developers leave and project is handed over to client.

CPLX	High	All possible end cases and exceptions should be thought of and tested for the application to be a success.
RUSE	Low	This module is built specifically for this project and can not be used elsewhere.
TIME	Nominal	This module will be used only before the game goes into production, and later on for testing and maintenance purposes only. The users do not see it.
STOR	Nominal	Testing doesn't require a lot of memory space.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Low	Team is new to Java spring, and has not done testing in that framework before.
LTEX	Nominal	Team has experience with Java.
PLEX	Low	Team doesn't have previous experience developing in Java spring.
TOOL	Very High	Tool used is IntelliJ Idea, MySQL and HiediSQL which are well established tools used in industries for development.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 19: COCOMOII Cost Driver, Module 10: Game Creation

Cost Driver	Value	Rationale
RELY	High	High level of risk if game creation fails. The game can not be played unless this module functions properly.
DATA	High	Module should be tested thoroughly for all types of end cases and invalid cases.
DOCU	High	This is one of the most important modules of the application and needs to be documented well before handing it over to the client.
CPLX	Nominal	The database and models needed to be designed from scratch, but business logic is fairly simple.
RUSE	Low	This module is built specifically for this project and can not be used elsewhere.
TIME	Nominal	This module will be used only at the start of each new game.
STOR	Nominal	Doesn't require a lot of memory space to store the details of a new game.
PVOL	Low	Platform volatility is expected to be low since technologies being used like MySQL, php, Java, HTML, etc. don't change frequently.
ACAP	High	Team members are all graduate students who have good analytical skills.
PCAP	High	Team members are graduate students with Computer Science background so have good programming skills.
PCON	Very High	Team members will work on project till the end of the semester.
APEX	Nominal	Team has experience with database and model creation..

LTEX	Nominal	Team has experience with SQL and Java.
PLEX	Nominal	Team doesn't have previous experience developing in Java spring, but is proficient with MySQL.
TOOL	Very High	Tool used is IntelliJ Idea, MySQL and HiediSQL which are well established tools used in industries for development.
SITE	Extra High	Most of the team members live near campus so it is easy to schedule meetings and collaborate with each other.
SCED	Nominal	The schedule is fixed for 12 weeks.

Based on the scale drivers and the cost drivers, the following is the COINCOMOII estimate of effort:

<div> <input type="button" value="Subcomponent"/> <input type="button" value="Subcomponent(s)"/> <input type="button" value="More Actions ..."/> </div>												
X	Name	Size	Labor Rate (\$/Month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	Risk
<input type="checkbox"/>	Login Page	168	0.0	0.48	High Level	0.55	0.26	637.19	0.00	0.00	0.0	0.0
<input type="checkbox"/>	Register Page	273	0.0	0.48	High Level	0.89	0.43	637.19	0.00	0.00	0.1	0.0
<input type="checkbox"/>	Game Page	1,122	0.0	1.27	High Level	3.66	4.67	240.43	0.00	0.00	0.6	1.9
<input type="checkbox"/>	Loading Page	100	0.0	0.22	High Level	0.33	0.07	1373.93	0.00	0.00	0.0	0.0
<input type="checkbox"/>	Profile Page	200	0.0	0.36	High Level	0.65	0.23	860.50	0.00	0.00	0.0	0.0
<input type="checkbox"/>	Leader Board	160	0.0	0.36	High Level	0.52	0.19	853.44	0.00	0.00	0.0	0.0
<input type="checkbox"/>	User Authentication	310	0.0	0.71	JAVA	1.01	0.72	430.74	0.00	0.00	0.1	0.0
<input type="checkbox"/>	Detonate	462	0.0	0.69	JAVA	1.51	1.03	447.43	0.00	0.00	0.1	0.0
<input type="checkbox"/>	Unit Tests	450	0.0	0.56	JAVA	1.47	0.82	546.31	0.00	0.00	0.1	0.0
<input type="checkbox"/>	Game Creation	420	0.0	0.43	JAVA	1.37	0.60	704.75	0.00	0.00	0.1	0.0

Estimation							
Total Lines Of Code: 3,665		Hours/PM: 152.0					
Estimated	Effort	Schedule	PROD	COST	INST	Staff	Risk
Optimistic	7.22	6.83	507.71	0.00	0.00	1.1	
Most Likely	9.02	7.32	406.17	0.00	0.00	1.2	1.9
Pessimistic	11.28	7.85	324.93	0.00	0.00	1.4	

Figure 1: COINCOMOII Estimate

Estimation:

Number of SLOC: 3,665

Effort needed (Pessimistic): 11.28 person-month

Schedule (Pessimistic): 7.85

Each Member Works: 18hrs/week for 12 weeks

Time Spent by Members:

A: 18 hrs/week × 7 members × 4 weeks = 504 hrs/month

B: 18 hrs/week × 7 members × 12 weeks = 1,512 total hours

Time Needed:

A: $11.28 \text{ person-month} \times 152 \text{ hrs/person-month} = 1,714.56 \text{ total hours}$

B: $(11.28 \text{ person-month} \times 152 \text{ hrs/person-month}) \div 504 = 3.4 \text{ months}$

Is it possible: Yes, according to the above estimation, the project will take 3.4 months to complete. The team has 12 weeks which is within the estimated time to complete the project.

6. Iteration Plan

6.1 Plan

There will be two cycles to the Development phase's construction iteration. The first cycle will aim at the completion of the core capabilities of the system and ensuring they work well in time for the Core Capability Drive (CCD). During the CCD, the client will test the system for each of the core capabilities he had specified and provide feedback on the same, which is vital as input to refine the system further in the next cycle and also useful for coming up with scenarios for end case testing.

The second cycle will be utilized for testing the full functionality of the system, fixing bugs and modifying the display and user experience depending on feedback from the client during the CCD and if required also build on the core capabilities to extend any particular feature.

After the second cycle, the team will focus on transitioning the system smoothly, documenting the technical aspects of the application and training the stakeholders on how to use, modify and manage the system.

6.1.1 Capabilities to be implemented

Given below are the core features to be implemented in the two cycles, and each of these have been documented in detail in the Operational Concept Description (OCD) and the System and Software Architecture Description (SSAD). The ID corresponds to the IDs of these features as specified in the OCD.

Table 20: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority	Iteration
1	OC-1	User Sign Up/Login: Users should be able to register and login to the system to play the game.	Must Have	1
2	OC-2	Gameplay based on Paytable Calculation: The points spent by the users to buy bombs to play the game depending on the multiplier chosen and	Must Have	1

		the calculation of the points won depending on the sequence of prize items won should be implemented based on the pay table calculation specified by the client.		
3	OC-3	Edit Profile: The users should be able to edit their own profile which includes first name, last name, email id, password and avatar.	Must Have	1
4	OC-4	Leaderboard: The top five high scorers list should be displayed to the users. This will give the players the competitive edge to try and beat the high scorers and be one themselves.	Should Have	1
5	LOS-1	Scalable System: The game should be designed such that it can be scaled in the future by the maintenance team without much difficulty.	Should Have	1
6	LOS-2	Responsiveness: The game should appear and function the same on laptops, tablets and mobile phones regardless of the type of browser used.	Must Have	1

6.1.2 Capabilities to be tested

Testing will be done throughout each iteration, as and when features are implemented. However, thorough testing will be carried out on the capabilities in the second cycle. This testing cycle is vital as it will help us uncover hidden bugs for end cases, or rainy day scenarios. Given below are the capabilities that will be tested during the second cycle; it includes all the capabilities mentioned in the section above. The Test Plan and Cases document contains detailed explanation of the same.

Table 21: Construction iteration capabilities to be tested

ID	Capability	Description	Priority	Iteration
1	OC-1	User Sign Up/Login: Users should be able to register and login to the system to play the game.	Must Have	1
2	OC-2	Gameplay based on Paytable Calculation: The points spent by the users to buy bombs to play the game depending on the multiplier chosen and	Must Have	1

		the calculation of the points won depending on the sequence of prize items won should be implemented based on the pay table calculation specified by the client.		
3	OC-3	Edit Profile: The users should be able to edit their own profile which includes first name, last name, email id, password.	Must Have	1
4	OC-4	Leaderboard: The top five high scorers list should be displayed to the users. This will give the players the competitive edge to try and beat the high scorers and be one themselves.	Should Have	1
5	LOS-1	Scalable System: The game should be designed such that it can be scaled in the future by the maintenance team without much difficulty.	Should Have	1
6	LOS-2	Responsiveness: The game should appear and function the same on laptops, tablets and mobile phones regardless of the type of browser used.	Must Have	1

6.1.3 Capabilities not to be tested

All the features and capabilities are to be tested before handing the project over to the client.

6.1.4 CCD Preparation Plans

The Core Capability Drive through (CCD) is scheduled on November 16th, 2016 at the SAL building, room SAL322. The people participating in the CCD are the client's (Crazy Cool Apps) representative Mr. Ankush Prasad and the development team.

During the CCD the client will test the basic functionalities of the application and his feedback will be noted down by the team so that it guides the changes to be made in the next iteration. Given below is the usage scenario supported by the application.

The client will act in the role of a new user who wants to play the Newlette Coins game. Here is a sequence of actions he needs to perform to test all the core functionalities:

1. Register as a new user to the game by providing user's information and a password.

2. Login to the game with the credentials of the newly created user. Also test the forgot password functionality of the login page by entering the email id and making sure that the user gets a temporary password to login and change his actual password.
3. While the game page is loading, make sure that the leaderboard is being displayed to the user with the top five scorers' names only.
4. Once the game page is loaded, the user clicks different cells on the board to make sure that the bombs are being placed correctly. The user should be able to place only 4 bombs as per the client's requirements. The amount of points spent by the user should be updated as and when he/she places the bombs on the board.
5. The detonate button should appear enabled only after the user has placed the 4 bombs. He should also be able to select different multipliers before clicking the detonate button.
6. Once the detonate button is clicked, the bombs should explode and the prize items hidden in those chosen cells should be visible along with the number of points won by the user. These points should get added to the user's total balance which is displayed on the top right corner of the screen.
7. The user should be able to see pop ups explaining the game to the user once he clicks on the pay table and help buttons.
8. The user should be able to see seamless animation for all the actions performed in the game and different game sounds should be present. On clicking the mute button, the game sounds should stop. On moving the volume control bar, the user should be able to experience appropriate changes in the game sounds.
9. Now that the major part of the game is tested, the user can click on the edit profile button to go to the edit profile page. There he should be able to make changes to his first name, last name, email id, password and avatar.
10. The user should be able to log off properly without any discrepancies in his total balance.
11. The user should try to play the game on a laptop, tablet and mobile phone so that he can test the functionality of the game on different types of devices and make sure it works as expected.

By performing the above mentioned actions, all the core features of the game will be tested by the client.

Given below are the development team's preparation plans for the CCD:

1. Make sure that all logged bugs are resolved & closed after verification before CCD.
2. Develop the game's core features well in advance to the CCD session and keep a buffer of a few days for testing only.
3. Ensure that the AWS server is up and running and that the server calls are functioning accurately.
4. Prepopulate the database with a few users so that the leader board can show top 5 scorers.
5. Create a feedback form (google form) to collect the client's feedback on the day of CCD.
6. Do a dry run with the whole team before the actual CCD session.
7. Ensure all of the client's requirements have been implemented and tested.

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

The table below specifies the capabilities that were implemented, the test cases that were performed on it and the results obtained for each of them.

Table 22: Capabilities implemented, tested, and results

ID	Capability	Test Case	Test Results	If fail, why?
OC-1	User Sign Up/Login	TC-04-01	Pass	
		TC-04-02	Pass	
		TC-04-03	Pass	
		TC-04-04	Pass	
		TC-04-05	Pass	
		TC-05-01	Pass	
		TC-05-02	Pass	
		TC-05-03	Pass	
		TC-05-04	Pass	
		TC-05-05	Pass	
OC-2	Gameplay based on Paytable Calculation	TC-01-01	Pass	
		TC-01-02	Pass	
		TC-01-03	Pass	
		TC-01-04	Pass	
		TC-01-05	Pass	
		TC-06-01	Pass	
		TC-06-02	Pass	
		TC-06-03	Pass	
		TC-06-04	Pass	
		TC-06-05	Pass	
		TC-07-01	Pass	
		TC-07-02	Pass	
		TC-07-03	Pass	
		TC-07-04	Pass	
		TC-07-05	Pass	
OC-3	Edit Profile	TC-08-01	Pass	
		TC-08-02	Pass	
		TC-08-03	Pass	
		TC-08-04	Pass	

OC-4	Leaderboard	TC-02-01	Pass	
		TC-02-02	Pass	
		TC-02-03	Pass	
		TC-02-04	Pass	
		TC-02-05	Pass	
LOS-2	Responsiveness	TC-03-01	Pass	
		TC-03-02	Pass	
		TC-03-03	Pass	
		TC-03-04	Pass	
		TC-03-05	Pass	

6.2.2 Core Capabilities Drive-Through Results

Positive Feedback:

1. Since the client has been testing the game hands-on for quite some time now, he was impressed to see the changes in the UI design and the color theme of the game.
2. The client liked the animations and the addition of the volume bar control.
3. All the changes the client had requested were implemented so he was more than satisfied.

Improvements Needed/Suggested:

1. When the user is out of balance, he/she shouldn't be able to place bombs on the board at all.
2. In the page that displays user's game history, the actual date and time (timestamp) should be displayed instead of the current message like "two days ago" etc.

Changes to-be Considered:

1. User avatar display should be implemented.

Risks:

1. Possible UI scaling issues on devices that the game is not tested on.

6.3 Adherence to Plan

The development went according to plan and the main features of the game was completed in time for the CCD session. The team strictly adhered to the project schedule. One improvement that can be made is to increase the amount of time spent on testing the game alongside development. All the requirements need to be thoroughly tested for end cases and rainy day scenarios. Another issue we should be careful about is merge conflicts when we push our code changes to github. We should test our local copy thoroughly before pushing it so as to avoid breaking the game, which we had encountered during development. This was however easily fixed that time by rolling back the latest changes made and then figuring out how the new changes broke the game. A fix was found and the development continued smoothly.