DUY TÂN
UNIVERSITY
INTERNATIONAL SCHOOL

GreenBig 5
Project Plan

Project Code: GB5

Document Code: GB5-PPD– v2.4

Mentor: Doctor. Habil. Binh, Nguyen Thanh

Group: C1SE.02

Chinh, Thai Huu
Chung, Hoang Bao
Hau, Phuc Bui
Loc, Nguyen Tien

Da Nang, 8-Sept-2021

SIGNATURE PAGE

Name	Signature	Date
Binh, Thanh Nguyen _____		_____31 - Nov- 2021

AUTHOR:	Chinh, Thai Huu Content Management	8-Sept-2021
REVIEWERS:	Chung, Bao Hoang Back-end developer	10-Sept-2021
	Hau, Phuc Bui Application Developer	10-Sept-2021
	Loc, Tien Nguyen Database Developer	10-Sept-2021
MENTOR:	Binh, Thanh Nguyen Stakeholder	31-Nov-2021

RECORD OF CHANGE

*A - Added M - Modified D – Deleted

Effective Date	Changed Item	A* M, D	Reason for Change	Revision Number
1	UI for Application	M	Improve UI for login in the app	
2	Create chatbot for Application	D	Not fit to the project	
3	Send question based on the scenario	A		
4	Manage content	A		
5	Manage indicator	A		
6	Develop register by OTP code	D	Out of budget	
7	Develop register interface	A	Change from OTP by traditional register	
8	Update physical database	A		
9	Manage GB5 scenario	A		
10	Update GB5 Scenario	M	Improve Scenario	
11	Update GB5 Dashboard	M	Upgrade System	

TABLE OF CONTENTS

1. PROJECT OVERVIEW

1.1. *Project Description*

Project code	**GB5**	**Contract type**	**Internal Project**
End-User	**Nominal user**		
Project Type	**Internal**	**Project Manager/ Scrum master**	**Chinh, Huu Thai**
Project Category	**Development**		
Application type	**Information System**		

1.2. *Scope and Purpose*

1.2.a Project Propose

The aim of this project is to build a GreenBig5 information system (GB5), i.e. GB5 App, database and GB5 Dashboard. By using GB5 App, user activity data can be collected and used to predict her/his Personality based on Big5 traits. Afterwards, the user's personality can be seen as one of main key factors, which have linking to environmental concerns/impacts. As a result, government authorities, enterprises, as well as users would have an overview of the environment and have a better solution to change user behaviour and to reduce and prevent it from the bad effect.

In this phase, we focus on collecting user's personality data so that the GB5 AI model can predict user's personality traits. This would help us in finding the connection between personality and environmental concern and be prepared for the next phase.

1.2.b Project Scope

In this project scope, we implement these features about user data, Big5 data, Big5 indicators, Big5 question

- ❖ *For the Application (GB5 Application):*
  - ➢ *Login/logout.*
  - ➢ *Sign In.*
  - ➢ *View question.*
  - ➢ *Answer question.*
  - ➢ *View the ranking.*
- ❖ *For the Database (GB5 Database):*
  - ➢ *Storage user's information.*
  - ➢ *Storage user's personality traits.*

6

> ➢ *Storage Big5 Indicator, Big5 Question.*
> ➢ *Send the Big5 Questions to the Application based on the scenario.*
- ❖ *For the Dashboard:*
  > ➢ *Visualize the user's Big5 personality traits.*
  > ➢ *Visualize data into a chart, ...*
  > ➢ *Send a question to the user to continue to predict user personality.*

Language:
- ❖ Vietnamese
- ❖ English

Duration:
- ❖ 17 weeks

1.3. *Assumptions and Constraints*

No	Description	Note
Assumptions		
1	The personality traits to environment concern would not be done in this phase	Scope
2	User's personality will be predicted in this phase	Propose
3	Customer reviewers will get seven days to approve a milestone document. If no comments are received within this time period, it will be considered as approved.	External Interfaces
4	The project support for Android and IOS operated system	Scope
Constraints		
1	Module A must be completed and delivered to customer before 09-Sep because customer has to demo to its end user by 11-Sep	Schedule
2	The project shall conform to security requirements specified by the customer in the NDA	Security

3	The product operated in high performance and have a page load of no more 10 seconds	Quality
4	The financial estimation for the project is at a budget limit of $4234	Budget
5	The project will be implemented by a team including 4 members	Resources

1.4. *Project Objectives*
1.4.1. Standard Objectives

Metrics	Unit	Committed	Note
Start Date	**dd-mmm-yy**	23-Aug-21	
End Date	**dd-mmm-yy**	12-Dec-21	
Duration	**days**	77 days	
Team Size	**4 Person(s)**	4 Persons	
Billable Effort /	**Person-day**	220	
Number of work hours per day for one engineer	**Person-hour**	4.5	

Table 1-1.4.1: *Resources*

Metric	Unit	Target SLS			Basic for Setting goal
		SLS	Average	USL	
Quality					
Customer Satisfaction	Point	8	9	9.5	Refer to Gx Target in the year 2020, 10% higher than previous project (A project)
Leakage	Wdef/UCP				
Process Compliance	NC/Ob				
Cost					
Effort Efficiency	%	80	75	90	
Correction Cost	%	65	60	75	
Delivery					
Timeliness	%	90	95	75	
Requirement Completeness	%	80	70	70	

Table 2-1.4.1: *Resources*

1.4.2. Specific Objectives
- Based on the human resources with allowable time and cost, we will build a system to predict user's personality
- This information system operated with high performance and safety for the user. User security data is encrypted and stored carefully, avoiding data loss.
- The deployment system minimizes defects and good control of risks by the project team.
- Strengthen brand promotion activities and bring products to users.
- Deploying applications will be operated quarterly for quick delivery to customer.

1.5. Critical Dependencies

No	Dependency	Expected delivery date	Note
1	**GB5 Application**	**20- Otc-2021**	
2	**GB5 Database**	**21-Otc-2021**	
3	**GB5 Dashboard**	**15-Nov-2021**	

1.6. Project Risk

Risk	Description	Probability	Impact	Mitigation Strategy
Incorrect requirements	Developing the product which does not accord with the requirements	3	5	Discuss and Communicate frequently with Stakeholders
Estimate working time	Actual working time is not enough to finish a task compared to the estimated previous time	1	3	Review old tasks and evaluations to estimate for the new task. Replan for each sprint.
People	Team member who is ill, has health problems, or busy	3	4	Notify the scrum master (or ask a colleague to help) Complete the assigned tasks when possible

Lack of technical experiences	Managing harmful content in the question is a difficult technique that all members need to research and develop.	4	5	Spend a lot of time learning and training as well creating a new standard.
Team Communication	Team members can conflict with each other while discussing	4	4	Conduct a meeting to share knowledge, experience and learning methods.

2. PROJECT DEVELOPMENT APPROACH

2.1. *Technical Process*

2.1.1. Reasons for selecting

To follow with today's technology evolution, we want a flexible and easy model to adapt with the change. Also, our project will update new features in the near future. So, our product would become more interactive and intelligent.

Because, our team has a modest number of members as well as little experience of development. Therefore, we can't avoid problems that arise in the software development stages and requirements can change to be more suitable. For the traditional process require a lot of experience, skills and high accuracy

2.1.2. Agile Methodology [1]

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

Agile software development is more than frameworks such as Scrum, Extreme Programming, or Feature-Driven Development (FDD).

Agile software development is more than practices such as pair programming, test-driven development, stand-ups, planning sessions, and sprints.

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it. When you approach software development in a particular manner, it's generally good to live by these values and principles and use them to help figure out the right things to do given your particular context.
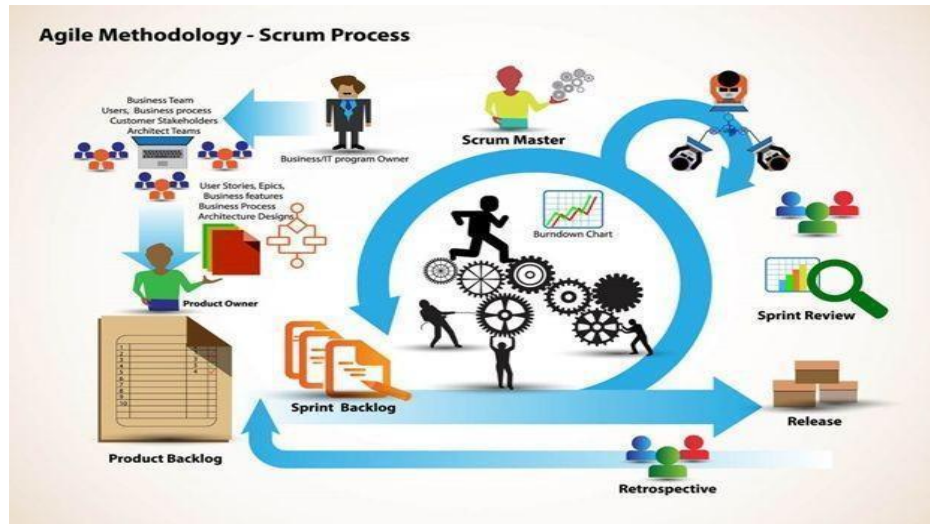
2.1.2.a.   Scrum Process

Figure 1-1.2.1.a: *Scrum process*

About Scrum:

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one [1].

Scrum is most often used to manage complex software and product development, using iterative and incremental practices. Scrum significantly increases productivity and reduces time to benefits relative to classic "waterfall" processes. Scrum processes enable organizations to adjust smoothly to rapidly-changing requirements and produce a product that meets evolving business goals.

An agile Scrum process benefits the organization by helping it to
+   Increase the quality of the deliverables
+   Cope better with change (and expect the changes)
+   Provide better estimates while spending less time creating them
+   Be more in control of the project schedule and state.

2.2. *Quality Management*

2.2.1. Estimates of Defects to be detected

Pre-release review defects

Process	Planned found by review	Actual found by review
Requirement	90	
<Work product>		
Design	80	
<Work product>		
Coding	100	
<Work product>		
Other	45	
<Work product>		
Total	315	

Table 1-2.1.1: *Pre-release review defects*

Pre-release test defects

Process	Planned found by review	Actual found by testing
Requirement	40	
<Work product>		
Design	35	
<Work product>		

Coding	150	
<Work product>		
Other	15	
<Work product>		
Total	**240**	

Table 1-2.1.1: *Pre-release test defect*

2.2.2. Strategy for Meeting Quality Objectives

Strategy	Expected Benefits
Do defect prevention using the standard defect prevention guidelines and process; use standards developed in Flutter/ Python for coding.	10–20% reduction in defect injection rate and about 2% improvement in productivity
Group review of program specs for first few/logically complex use cases. Group review of design docs/first time-generated code by project leader, developer, and one consultant.	Improvement in quality as overall defect removal efficiency will improve; some benefits in productivity as defects will be detected early
Introduction of RUP methodology and implementing the project in iterations. Milestone analysis and defect prevention exercise will be done after each Iteration.	Approximately 5% reduction in defect injection rate and 1% improvement in overall productivity

2.2.3. Quality Control

Review Item				Type of Review	Reviewer	When
Proposal				Group review	Binh, Thanh Nguyen	Initial
Project **plan** **Project** **schedule** **CM Plan**				Group review	Binh, Thanh Nguyen Loc, Tien Nguyen Chung, Bao Hoang Hau, Phuc Bui Chinh, Huu Thai	End of Initiation stage
Business analysis and requirements specification document, Use Case catalog				Group review	Loc, Tien Nguyen Chung, Bao Hoang Hau, Phuc Bui Chinh, Huu Thai	End of 70% requirement
Design document, object model				Group review	Loc, Tien Nguyen Chung, Bao Hoang Hau, Phuc Bui Chinh, Huu Thai	End of 90% design
Stage plans				One-person review	Binh, Thanh Nguyen	Beginning of each stage
Complex/ first specs incl. diagrams	**Time test**	**Generated cases,**	**Program interactive**	Group review	Binh, Thang Nguyen Chinh, Huu Thai Chung, Hoang Bao Loc, Tien Nguyen Hau, Phuc Bui	End of detailed design

Code	Group review	Chinh, Huu Thai Chung, Hoang Bao Loc, Tien Nguyen Hau, Phuc Bui	After coding for first few programs

2.2.4. Measurements Program

Data to be collected	Purpose	Responsible	When
Size: No. of KLOC// FP	Early estimate project cost	**PM/SM**	At the end of stages
Effort: No. person-day	Calculate project effort for scheduling	**Team members**	Daily
Quality: No. defects detected	Early evaluate product quality and the feasibility of the project	**Reviewer, Tester**	Right after the review/test
Schedule	Divide work and allocate resources properly, ensure the project is completed on time and on budget	**PM/SM**	Weekly and at the end of stages

2.3. *Unit Testing Strategy*

- *Grey Box:*
    - It is a combination of a Black Box and White Box testing. It is the type of testing in which the tester is aware of the internal functionality of a method or unit but not in

16

a deeper level like white box testing. In this, the user is partially aware of the internal functionality of a system.
- Write test cases before fixing the defect and independent of each other.
- Write cases to verify behavior, also write test cases to ensure the performance of the code
- Execute test cases continuously and frequently.
- Using tool: Install and run Jest for writing unit test in NodeJS
- Isolation of a code – Isolate function to test it more rigorously. Isolate code to do Automated Unit Testing in a better way. Isolating functions/code helps to do testing in a good way. It helps to reveal dependencies between functions of code.

2.4. Integration Testing Strategy

- *Bottom up Strategy:*
  - The components below are first written and these are integrated first. The integration happens from bottom to top. If the calling component is yet to be developed, it is replaced by a specially written component called a Drive
  - When we finish each product backlog, we test it out before we finish.

- *Bigbang Strategy:*
  - All components are put together at the same time, there is no order, except all are integrated at the same time.
  - Towards the end of the project, we started to apply this tactic to test the entire application.

2.5. System Testing Strategy
- *Customer testing(Beta testing) strategy:*
  - Beta testing is a type of user acceptance testing where the product team gives a nearly finished product to a group of target users to evaluate product performance in the real world.
  - We are rolling out a beta app on the Google Store early on for testing. After that, we gathered all the feedback and improved our system.
3. ESTIMATION
3.1. Size
Total number of FP: 68

Software Scale Drivers	
Precedentedness	*Nominal*
Development Flexibility	*Nominal*

Architecture / Risk Resolution	*Nominal*
Team Cohesion	*Very High*
Process Maturity	*Nominal*

The Size estimation is documented in Page 16-17

Software Cost Drivers				
Product			**Personnel**	
Required Software Reliability	*Nominal*		**Analyst Capability**	*High*
Database Size	*Nominal*		**Programmer Capability**	*High*
Product Complexity	*Nominal*		**Personnel Continuity**	*Nominal*
Developed for Reusability	*High*		**Application Experience**	*High*
Documentation Match to Lifecycle Needs	*Nominal*		**Platform Experience**	*High*
			Language and Toolset Experience	*High*
Project			**Platform**	
Use of Software Tools	*High*		**Time Constraint**	*Nominal*
Development	*Nominal*		**Storage Constraint**	*Nominal*
Required Development Schedule	*Nominal*		**Platform Volatility**	*Nominal*

Software Development (Elaboration and Construction)

> **Effort = 9.6 Person-months**
> **Schedule = 7.7 Months**
> **Cost = \$2873**
> **Total Equivalent Size = 5440 SLOC**
> **Effort Adjustment Factor (EAF) = 0.52**

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.6	1.0	0.6	$172
Elaboration	2.3	2.9	0.8	$690
Construction	7.3	4.8	1.5	$2184
Transition	1.1	1.0	1.2	$345

3.2. *Effort*
The Effort estimation is documented in page 18-19

Activity/Process	Total budgeted Effort Usage (pd)	Total % budgeted Effort Usage (%)	Sprint 1		Sprint 2		Sprint 3		Sprint 4	
			No	%	No	%	No	%	No	%
Requirement	25	10.5	8	21.1	4	10.5	3	7.	2	5.9
Design	11	5.0	3	7.9	2	5.3	2	5	2	5.9
Coding	100	40.0	10	26.3	1	36.8	1	45	1	47.1
Unit Testing	11	5.0	0	0.0	2	5.3	3	7.	2	5.9
Testing	22	10.0	3	7.9	4	10.5	4	10	4	11.8

Deployment	11	5.0	0	0.0	2	5.3	2	5	2	5.9
Support for Acceptance Test	10	4.1	0	0.0	2	5.3	1	2.	1	2.9
Project Planning	9	4.1	4	10.5	1	2.6	1	2.	1	2.9
Project monitoring	14	6.4	3	7.9	2	5.3	3	7.	2	5.9
Quality Assurance	14	6.4	2	5.3	2	5.3	3	7.	2	5.9
Training	8	3.6	5	13.2	3	7.9	0	0	0	0

Total	235	100	38	100	38	100	40	100	34	10	3	10	3	100

3.3. Schedule

3.3.1. Project Milestone & Deliverables

1. Deployment GB5 Scenario
2. Deployment Application
3. Deployment Database
4. Deployment Dashboard

3.3.2.   Work Breakdown Structure
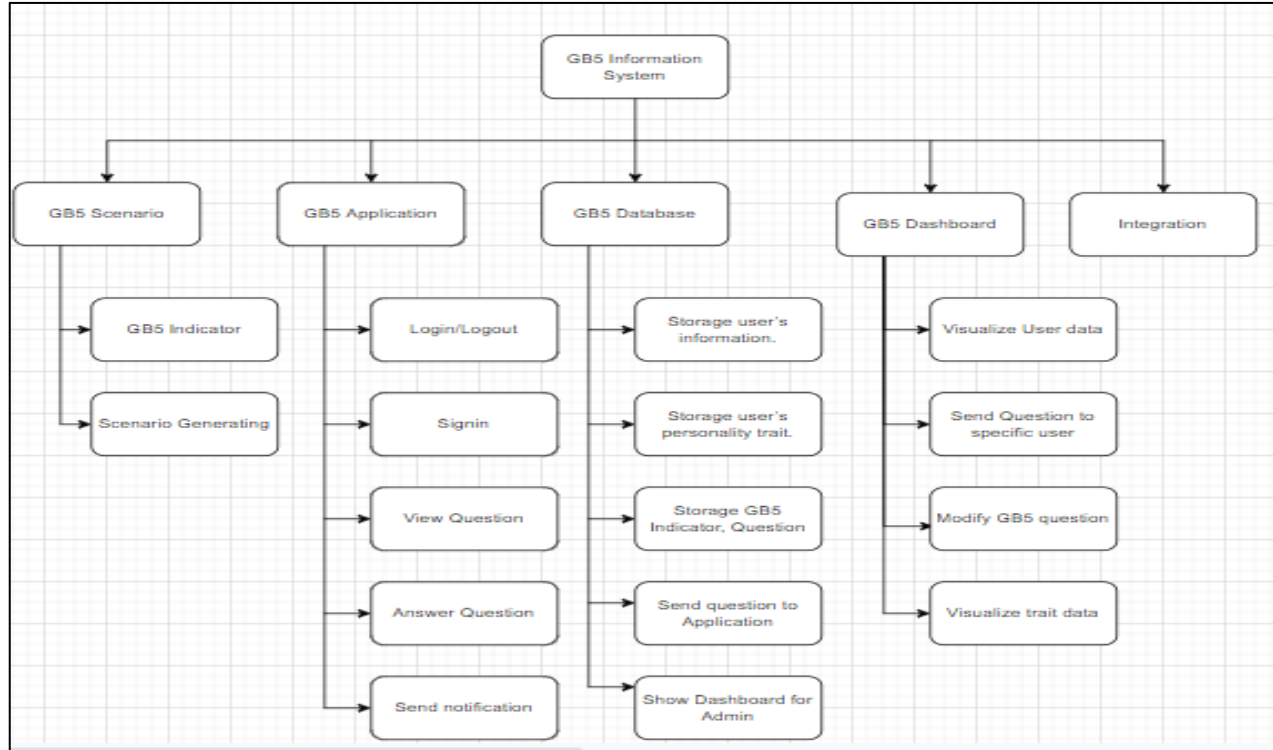
Table 3.3.2: *WBS*

3.3.3. Detailed Schedule

WBS	Task	Duration(s)	Start	End	Assign to
1	**Initial**	**7**	**15Aug 2021**	**22 Aug 2021**	**Team, Mentor**
1.1	Project's Kick-off Meeting	**1**	**15 Aug 2021**	**16 Aug 2021**	**Team, Mentor**
1.2	Collect and analyse requirements	**4**	**16 Aug 2021**	**20 Aug 2021**	**Team, Mentor**
1.3	Setup Development Environment	**1**	**20 Aug 2021**	**21 Aug 2021**	**Team, Mentor**

1.4	Research Technical	1	21 Aug 2021	22 Aug 2021	Team, Mentor
2	**Develop**	**105**	**23 Aug 2021**	**15 Dec 2021**	**Team**
2.1	**Sprint 1**	**28**	**23 Aug 2021**	**20 Sep 2021**	**Team**
1	Collect, reconstruct Big-five indicators[*] and manage content for Big-five questions.				**Team**
2	Documentation about GB5 Application functional.				**Team**
3	Documentation about GB5 database.				**Team**
4	Design GB5 Database				**Team**
5	Build GB5 Application UI.				**Team**
6	Testing.				**Team**
2.1	**Sprint 2**	**28**	**22 Sep 2021**	**20 Oct 2021**	**Team**
1	Continue collect Big-five indicator				**Team**
2	Develop GB5 Application.				**Team**
3	Inspect GB5 question data.				**Team**
4	Develop GB5 Application.				**Team**
5	Develop GB5 database.				**Team**

6	Design GB5 Dashboard				**Team**
2.3	**Sprint 3**	**28**	**23 Oct 2021**	**21 Nov 2021**	**Team**
1	Develop GB5 Application				**Team**
2	Design GB5 Dashboard UI.				**Team**
3	Develop GB5 Dashboard				**Team**
4	Testing				**Team**
2.4	**Sprint 4**	**21**	**24 Nov 2021**	**15 Dec 2021**	**Team**
1	Develop GB5 Dashboard				**Team**
2	Visualize user data.				**Team**
3	Analyze personality data.				**Team**
4	Documentation about user's personality data.				**Team**
5	**Analyze personality data and correlation to pro-environmental behavior.**				**Team**

3.3.4. Project Schedule

The detailed project schedule is available here The Project Schedule is weekly updated by the Project Manager.

No.	Activity	Start date	Responsible	Note
Defect Prevention				
	Task 1			
	Task 2			
Quality Control				
	Review: Work Product 1			
	Review: Work Product 2			
	Review: Work Product 3			
Project Tracking				
	<Stage name> milestone review meeting			
	<Stage name> milestone review meeting			
Configuration Management				
	<Baseline Name>			
	<Baseline Name>			
QA				

	Final Inspection: Deliverable 1			
	Final Inspection: Deliverable 2			
	Baseline audit: Startup			
	Baseline audit: Wrap-up			

3.4. Resource

Specified as in the section _Project Team_

3.5. Infrastructure

Work/Product	Purpose	Expected Availability by	Note
Development Environment			
NT Server	Operating System	Initiation stage	
IOS	Operating System		
Android	Operating System		
Database	MongoDB		
Flutter	Development language for Application		
Python	Development language for AI model		
NodeJS	Development Environment		
Hardware & Software			

1GB space on server			
Pycharm	Development		
Android Studio	Development		
MongoDB	Database		
Rational Rose	Design		
Other Tools			
CVS	Source version control	Definition stage	
Nunit	Unit Test	Construction stage	
DMS	Defect logging and tracking	Definition stage	
Timesheet	Effort logging	Initiation stage	
FI	Project management tool	Initiation stage	
MS Project	Task tracking	Initiation stage	

3.6. *Training Plan*

Training Area	Participants	When, Duration	Waiver Criteria
Technical			
Python Language	Chung,Chinh	17 days	If already trained
Flutter Framework	Loc, Hau	8 days	If already trained
Process			

Quality system	All member	3 hrs	Mandatory
Configuration management	All member	2 hrs	If already trained for CC. For others, on-the- job training
Group review	All member	4 hrs	If already trained
Defect prevention	All member	4.5 hrs	Mandatory
SPC tool	All member	4.5 hrs	If already trained
RUP methodology	All member	2 hrs	Mandatory

3.7. *Finance*

Item	Total Budget	% Budget	Budget in Period											Note
			W1 - Sep	W2 - Sep	W3 - Sep	W4 - Sep	W1 - Oct	W2 - Oct	W3 - Oct	W4 - Oct	W1 - Nov	W2 - Nov	W3 - No v	
Purchases (COTS)	**150**	**15.6**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**0**
Team building	**200**	**20.8**	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**3**	**0**	**0**
Tools	**100**	**10.4**	**8**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Travel costs	130	13.5	1	1	1	1	1	1	1	1	1	1	10	10
Training	80	8.3	3	5	0	0	0	0	0	0	0	0	0	0
Review activities	250	26.0	1	2	1	2	1	2	1	2	1	2	15	20
Other	50	5.2	0	0	0	0	0	0	0	0	0	0	0	0
Total	**960**	**100**	1	1	2	8	2	6	2	8	2	6	25	30

4. PROJECT ORGANIZATION
4.1. Organization Structure

Scrum Master	- Communicate the value of Scrum - Teach the organization on Scrum to maximize business value - Preserve the integrity and spirit of the Scrum framework - Serve as a coach and mentor to members of the Team - Respectfully hold the Team, Product Owner and Stakeholders accountable for their commitments - Continually work with the Team and business to find and implement improvements - As a timekeeper - Helping the team agree on what they can achieve during each development sprint (or other period of time).	Chinh, Thai Huu

	• Facilitating the daily standup (sometimes called the daily scrum) and helping the team reach consensus on each of the three questions. • Helping the team continuously make progress on the project by making sure each person is working on the right tasks, helping to remove any obstacles to the team members' progress, and protecting the team from distractions.	
Product Owner	• A spokesperson for the customer and needs to represent them • Gathers, manages, and prioritizes the product backlog. • Has technical product knowledge or specific domain expertise. • Tracks progress towards the release of a product.	Chung, Hoang Bao
Developer	• Responsible for quality • Responsible for delivering the potentially shippable product of the Application each sprint • Report progress based on the remaining time • Self-organized • Owns the Sprint backlog	All members
Mentor	• Guide on the process. • Monitoring all activities of the Team. • Help with anything. • Reviews project documents • Reviews product	Binh, Nguyen Thanh

4.2. Project Team

Full Name	Position
Binh, Nguyen Thanh	Mentor
Chinh, Thai Huu	Scrum Master, Dev-team
Chung, Hoang Bao	Product Owner, Dev-team
Loc, Nguyen Tien	Dev-team
Hau, Bui Phuc	Dev-team

5. COMMUNICATION & REPORTING

Communication Type	Method / Tool	When	Inform ation	Participants / Responsible
Project Task Tracking				
Task scheduling	MS Project Trello	At the beginning of every stage, and weekly Refinement and rescheduling as necessary		Project Mgr(s)
Task assignment	Trello	Weekly		Project Tech leader nical
Project Meeting				
Kick-off Meeting	Face to face Google meet Slack	Initiation stage	Project introduction; Project plan review; Risk identification; Obtainment of commitment of relevant stakeholders	Project Mgr(s), Project Senior Manager, Project Team Members, QA
Project Progress Review Meetings	Face to face Google meet Slack	Weekly event and on	Communicate project status Communicate and resolve any open issue, risks, and	Project Mgr(s), Project Team Members

			changes Discuss any suggested improvement	
Milestone Meetings	Face to face Google meet Slack	Before milestones	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	Project Mgr(s), Project Senior Manager, Project Team Members, QA

Communication Type	Method / Tool	When	Information	Partic ipants / Respo nsible
Project Post-mortem Meeting	Face to face Google meet Slack	Termination stage	Wrap-up Evaluate project performance; Team performance; share experiences	Project Mgr(s), Project Senior Manager, Project Team Members , QA

Transfer/Sharing of project documentation/information	Google Drive Google meet	When available	All project documentation and information	Project Mgr(s) Project Team Members , QA
Customer Communication and Reporting:				
Project Report	Agreed standard format between company and customer	5pm Monday, Weekly	Project status report, Issue requiring clarifications, escalation, if any	Project Manager Sub-Project Managers
Project Meetings with customer	Teleconference /TV Meeting	2pm Tuesday, Weekly	As above	Project Manager
Requirement gathering/clarification	Face to face Meeting Google meet	During requirement analysis phase	As in Q&A list	Project Manager Business Analyst
Communication with Senior Management				
Review Project Plan & Project schedule	Slack	Significant changes to WO, PP and Project schedule		Project Mgr

			(scope, objectives Organizatio n,HR, major milestone, deliverables)		
Project Progress Review	Slack	Weekly	Project status report, Issue requiring clarifications, escalation, if any	Project Mgr
Project Milestone Review	Google meet Slack	End of every stage	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	Project Mgr
Other Communication and Reporting:					
Raise issue or request service/support of BA groups (IT, Admin, QA, HR, Training, Recruitment,etc)	Calllog; phone; Slack	Upon request	Request content, expected completion date	Project Manager

6. CONFIGURATION MANAGEMENT

<Refer to the CM plan or insert here the contents of the CM plan as appropriated>

7. SECURITY ASPECTS

- The credential data is carefully secured by multi-layer encryption and data integrity is ensured. Regularly backup system data.
- Research on network attack prevention solutions to ensure data security, avoid being exploited and stolen data by hackers.
- Deploy project architecture with a high priority in security. Optimized architectural solutions enable the deployment of data security with 99% reliability.
- Social media, sharing and use of data must be approved by the end user and verified by the organization's management.

REFERENCES

No	Reference item	Issued Date	Source	Note
1	Agile Scrum	15-Aug-2021	https://www.atlassian.com/agile	
			https://www.cprime.com/resources/what-is-agile-what-is-scrum/	
			https://www.agilealliance.org/agile101/	
			The Scrum Framework by International Scrum Institute	
2	COCOMO II	15-Aug 2021	https://www.rose-hulman.edu/class/csse/csse372/201410/SlidePDFs/session12.pdf	
3	Software Standards	20-Aug 2021	https://www.nws.noaa.gov/oh/hrl/developers_docs/General_Software_Standards.pdf	
			https://standards.ieee.org/standard/12208-2017.html	

			https://sw-eng.larc.nasa.gov/	

DEFINITIONS AND ACRONYMS

Acronym	Definition	Note
PM	Project Manager	
PTL	Project Technical Leader	
QA	Quality Assurance Officer	
CC	Infrastructure Configuration Controller	
DV	Developer	
URD	User Requirement Document	
SRS	Software Requirement Specification	
ADD	Architecture Design Document	
DDD	Detail Design Document	
TP	Test Plan	
TC	Test Case	
SC	Source Code	
CM	Configuration Management	

CSCI	Computer Software Configuration Items	
CI	Configuration Item	
CCB	Change Control Board	
GB5	Green Big 5	