Software Project Management Plan

Team 5

January 3, 2021

Team Members

Bunnarith Heang
Vichea Heng
Sovath Chean
Seakmeng Chheang

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	03/Jan/21	Initial draft
V2.0	5/Jan/21	Final Draft

Document Storage

This document is stored at:
https://github.com/Notato-SE/documents

Document Owner

Seakmeng Chheang is responsible for developing and maintaining this document.

Table of Contents

1. Overview

1.1. Purpose and Scope

Team 5 is interested in creating Notato, a daily-need web application, which has a simple calculator, scientific calculator, converter, and randomizer. Notator also provides save randomizing list functionality to make users' life easier. It combines the power of backend and API service (3-rd party) to be able to run a smooth operation regardless of device specification. Users will be able to notice the really low power consumption of this app over the time.

Notato will be developed with Laravel for backend and VueJS for frontend with the help of 3rd party API for the conversion feature.

We do not only build the project for the final presentation but also to contribute a useful web application to the needy people out there. Hence, we will open source this project, as long as it's free to host the projects, both frontend and backend for modification or inspection for learning.

1.2. Goals and Objectives

1.2.1. Goals

- Create a functional app with well-designed documentation, design and features.
- Learn how to work with a standard software development life cycle
- Deliver an app for the final project

1.2.2. Objectives

- Create a web application for general purpose users
- Host the source code both frontend and backend to the programming community

1.3. Project Deliverables

Date	Deliverable
18/Dec/2020	- A draft requirement document
26/Dec/2020	- Use case diagram
02/Jan/2021	- User stories - cross-functional diagram - Wireframe UI in Adobe XD - SDA v.1 - SRS v.1 - SPMP v.1

8/Jan/2021	- SDA v.2 - SRS v.2 - SPMP v.2
15/Jan/2021	- Simple and Scientific Calculator
22/Jan/2021	- Authentication - Some features of Randomizer
29/Jan/2021	- Save randomized list - Export randomized list to excel

1.4. Assumptions and Constraints

1.4.1. Assumptions

- All team members will be able to complete tasks on time
- Any issue will be raised during the meeting

1.4.2. Constraints

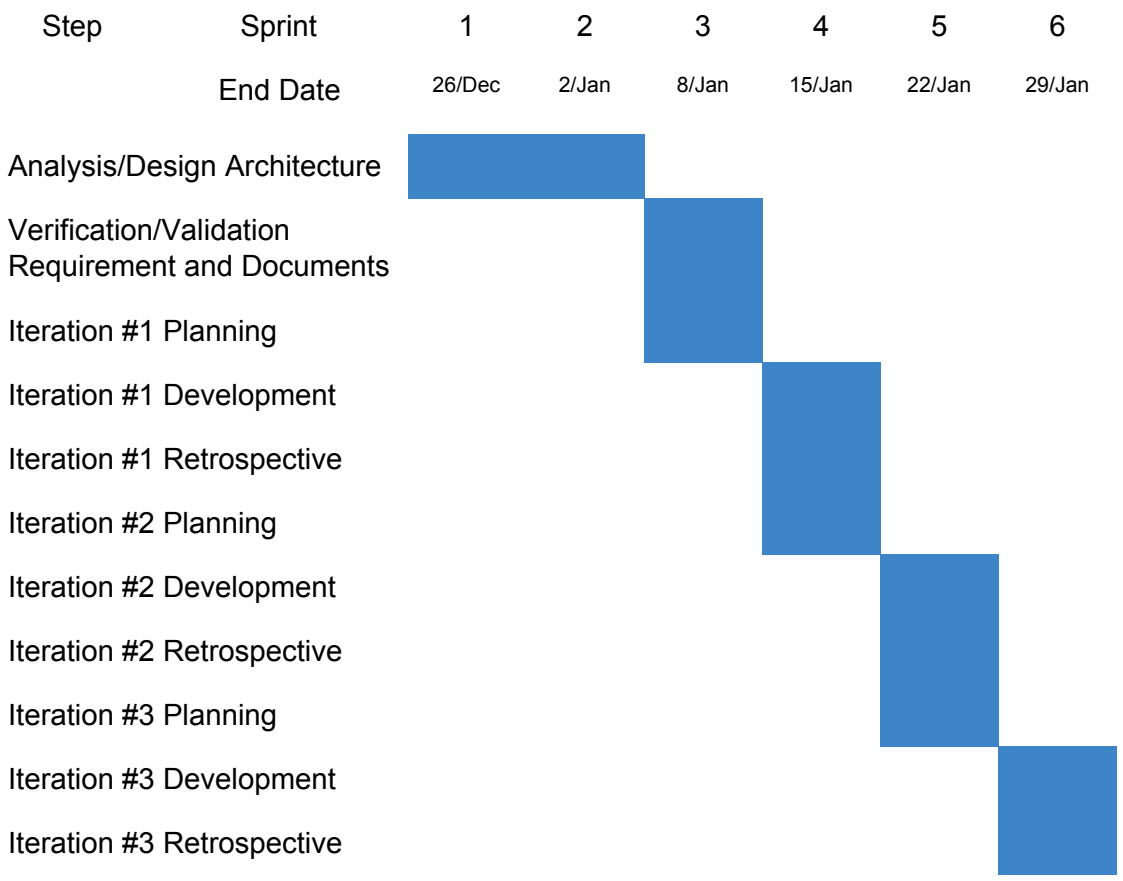
- 3-rd Party API service is used in conversion which we cannot guarantee the reliability and accuracy
- Users need a browser with javascript-enabled to run this web application
- Users need an internet connection to first load this web application

1.5. Schedule and Budget

1.5.1. Cost Estimate

- 4 software engineers each 4 story points per week
  4 * 4 = 16 story points/per week
- 6 sprints totally
  6 * 16 = 96 story points in total to complete the project

1.5.2. Schedule Summary



1.6. Success Criteria

All team members are equipped with skill on their own fields with a modern interface allowing users to work seamlessly with the application.

1.7. Definitions

Term	Definition
Notato	A simple web application with calculator, converter and randomizer.
Story Point	Story point is the estimated efforts for the team; 1 story point equivalent to 1 hour of work
Users	Refers to people interact with the web app, whether registered or not

2. Startup Plan

2.1. Team Organization

Role	Actor(s)	Responsibility
Team Lead	Seakmeng	Call for meetings, organize projects and distribute tasks to everyone.
Backend Developer	Sovath, Vichea, and Bunnarith	Implements Authentication and Randomizer.
Frontend Developer	Seakmeng	Design UI/UX and integrate API.
Requirement Engineer	Sovath, Vichea, Bunnarith and Seakmeng	Drafting requirement, design use case diagram and user stories.
DevOps	Vichea, and Bunnarith	Setup DigitalOcean to host a project, configure the server, and maintenance the server

2.2. Project Communications

Event	Info	Audience	Format	Frequency
Standup	Perform in the middle of the sprint to discuss the issue that we faced during the sprint	All team members	Discord/On-Campus Meeting	Once per sprint
Backlog Grooming	Perform if there any change requests from stakeholders	All team members	Discord/On-Campus Meeting	
Retrospective and Sprint	Perform at the end of each sprint	All team members	Discord/On-Campus Meeting	Once per sprint

Planning	to discuss what went well or wrong, then planning for the next sprint			
Project status	Keep track of team velocity	All team members	Jira	
Small discussion	Resolve any small concern, issue	All team members	Telegram	

2.3. Technical Process

Scrum + Incremental Software Development Process will be used as a method to develop this application.

- Scrum ceremonies such as Retrospective and Planning will be celebrated at the same time at the end of each sprint.
- Feedback, on what went well and what went wrong, from all team members will be collected for improvement.
- Backlog Grooming will be done before Retrospective and Planning, if there are change requests from a stakeholder. Change requests will be taken into analysis, discussion and added to the product backlog if all team members acknowledged and agreed.

2.4. Tools

- Laravel for backend
- VueJS for frontend
- Version control with git
- VSCode for code editor
- DigitalOcean for hosting

3. Work Plan

3.1. Resource Estimate

Team estimated effort is available here in Jira.

3.2. Release Plan

3.2.1. Iteration #1

- Frontend: Implement Simple and Scientific Calculator
- Backend: Authentication (register, login, reset password, change email and change password) and basic features of Randomizer (picker, team generator, and custom list).

3.2.2. Iteration #2

- Frontend: Implement Authentication (register, login, reset password, change email and change password) and basic features of Randomizer (picker, team generator, and custom list).
- Backend: Ability to save the data to the user account and export to excel.

3.2.3. Iteration #3

- Frontend: Ability to save the data and export to excel.
- Backend: N/A
- Devops: Hosting backend and frontend to the cloud (DigitalOcean).

4. Control Plan

4.1. Monitoring and Control

Milestones are included to reference where the project is scheduled to delivered:

4.2. Configuration Management Plan

1. All project works are stored in a Github organization dedicated for this project. Source code and documents are in separate repositories.
2. Requirement changing procedure:
    a. Raised the statement in the telegram group with what to change, reason to change, how change will be implemented, pros and cons of the change, risks and backup plan when the change is failed to implement.
    b. Team lead call for a meeting in discord to discuss about that
    c. Analyze the impact to the system architecture
    d. Decide whether to accept the change by voting system based on votes.
    e. Submit the change to a stakeholder (Mr. Neil Ian Uy)

5. Supporting Process Plans

5.1. Risk Management Plan

Rank	Risk	Possibility of Loss	Size of Loss	Risk Exposure	Response
1	3-rd Party API Service	Likely	Major	High	Mitigate: hosting our own API Service for the purpose of this project
2	Learning curve of new technology stack	Likely	Moderate	Moderate	Mitigate: Asking another team members who worked with that technology before
3	Schedule /Time line delivery	Unlikely	Major	High	Mitigate: daily standup when meeting on campus
4	Unknown constraint of new technology stack	Unlikely	Moderate	Moderate	Mitigate: prototype possible constraint before adding new feature to backlog

5.2. Test Plan

We will enforce TDD (Test-Driven Development) which encourages all team members to write the test first before writing the code to ensure each feature does what it's supposed to be done. Features without tests will not be allowed to merge into the project.

5.3. Product Acceptance Plan

Product will be tested by all team members after each sprint to ensure the behavior meets the requirement.