VotaFun

[Project Plan]
Version 1.2

By: **Team VotaFun**

Revision History

Revision Number	Date	Primary Author	Comments
1.0	27/09/2023	Ryan Teo Cher Kean	Initial draft
1.1	2/10/2023	Roy Lau Run-Xuan	Updated Section 5
1.2	9/10/2023	Ng Yue Jie Alphaeus	Updated Section 4

Table of Contents

Revision History 1

Table of Contents 2

1. Introduction 3

1.1. Project Overview 3

1.2. Project Description and Scope 3

1.2.1. Objectives 3

2. Project Organization 5

2.1. Team Structure 5

2.2. Roles and Responsibilities 5

2.3. Team Communication 6

3. Process Definition 7

3.1. Lifecycle Model 7

4. Schedule 8

4.1. Activity Dependencies and Schedule 8

4.2. Work Breakdown Structure 9

4.3. Work Packages 10

4.4. Activity Dependencies 10

4.5. Work Package Details 11

5. Project Estimates 16

5.1. Code Size Estimation using Function Points 16

5.1.1. Unadjusted Function Points (FP) 16

5.1.2. Adjusted Function Points 18

5.1.3. Lines of Code (LOC) 19

5.2. Efforts, Duration and Team Size Estimation 19

5.2.1. Distribution of Effort 20

5.3. Cost Estimates 20

6. Product Checklist 22

7. Best Practice Checklist 23

8. Risk Management 24

9. Quality Assurance 26

10. Monitoring & Control 27

1. Introduction

1.1. Project Overview

The project at hand is driven by the imperative need to improve decision-making efficiency among groups of friends when planning activities. The current situation often leads to time-consuming discussions and frustration. Our proposed design centres on the creation of a user-friendly platform while incorporating technology, specifically the integration of ChatGPT to facilitate structured discussions, activity proposals, and voting, all geared toward streamlining the decision-making process. This endeavour seeks to simplify communication by consolidating information within a dedicated platform, reducing misunderstandings and enhancing clarity. Moreover, the project aims to foster collaboration and consensus among friends with varying preferences, ultimately strengthening the overall decision-making process. Through these strategic design objectives, we aspire to revolutionise the way friends plan and decide on group activities, making it a more efficient and harmonious experience.

1.2. Project Description and Scope

In the context of various activities and choices, groups of friends face a recurring problem: they struggle to agree on a common activity due to their differing preferences and opinions. This diversity complicates decision-making, potentially leading to lengthy discussions and the inability to finalise plans efficiently.

The issue of indecision has real-world consequences. It results in inefficient use of time and undermines group cohesion. Prolonged debates can overshadow the enjoyment of shared activities, causing frustration and tension among friends. The initial enthusiasm and mutual respect can give way to disagreements and disrupt the group's harmonious dynamics.

A solution that balances individual preferences while streamlining decision-making is required. The goal of addressing these challenges is to restore efficiency to the process while maintaining the spirit of unity. Finding a method that respects individual input while allowing for faster decisions is critical for efficient planning.

1.2.1. Objectives

This document proposes a web application, VotaFun. We designed for groups of people in mind, particularly those looking to plan activities such as outdoor excursions, lunch plans or even movies to watch. The following are objectives that VotaFun should achieve:

Enhancing Decision-Making Efficiency

The project's primary goal is to improve decision-making among friends when planning group activities. The current situation of lengthy discussions and indecision frequently results in wasted time and frustration. Our goal in creating a user-friendly platform is to provide a structured environment where friends can collaboratively engage in discussions, propose activity ideas, and cast votes to streamline decision-making.

Simplifying Communication

Current communication channels are fragmented. Groups may discuss activities on different platforms, resulting in a sprawl of information. This can lead to misunderstanding since information is discontinuous between users. Our goal is to improve communication by providing a platform dedicated solely to activity planning. This platform will consolidate information, proposals, discussions, and votes, making it easier for friends to stay informed and involved in decision-making.

Collaboration in Promoting Consensus

We want to create a culture that values collaboration and compromise. Friends frequently have opposing viewpoints, making it difficult to reach an agreement. The project allows members in a group to vote anonymously. This encourages users to input their opinions when deciding on an activity when they might not have done so in an in-person setting. The app can then reach an activity which can best fit the group’s interests and can keep most of them happy.

2. Project Organization

2.1. Team Structure

The following is the list of executive roles, as required by CMM level 3.

- **Project Manager**: Ng Yue Jie Alphaeus
- **Lead Front-end Developer**: Abdul Siddiq Bin Mohd Yussaini
- **Front-end Developer**: Ryan Teo Cher Kean, Wang Xin Yan Lloyd
- **Lead Back-end Developer**: Roy Lau Run-Xuan
- **Back-end Developer**: Ng Yue Jie Alphaeus, Tran Trung Dung (Charles)
- **Quality Assurance Engineer / Manager**: Tran Trung Dung (Charles), Abdul Siddiq Bin Mohd Yussaini
- **Release Engineer / Manager**: Wang Xin Yan Lloyd

2.2. Roles and Responsibilities

Project Manager:

- The overall leader of the project team.
- Develop project plan, project proposal and project schedule.
- Establish and enforce quality standards for project deliverables.
- Communication bridge between the client and the project team.

Lead Front-end Developer:

- Design and architecture frontend components.
- Design user-friendly and intuitive frontend components.
- Implement and maintain testing processes for front-end components.
- Ensure code quality and conduct code reviews.

Front-end Developer:

- Translate design mockups into functional interfaces using CSS and Javascript.
- Write clean and maintainable code in Javascript using the React framework.
- Conduct tests such as unit, integration, and system testing for front-end components.

Lead Back-end Developer:

- Design and architecture of back-end components.
- Work with the frontend team to ensure seamless integration between the front and backend.
- Implement and maintain testing processes for backend components.
- Ensure code quality and conduct code reviews.

Back-end Developer:

- Develop and maintain sockets for frontend and backend communication.
- Write clean and maintainable code in Python using the Flask framework.
- Conduct tests such as unit, integration, and system testing for backend components.

Quality Assurance (QA) Engineer/ Manager:

- Develop test plans for the front-end and back-end teams to validate functionality and performance.
- Identify, document and track bugs and defects.
- Collaborate with the front-end and back-end teams to ensure the quality of CSS, Javascript and Python scripts.

Release Engineer / Manager:

- Responsible for planning, coordinating and delivering software releases.
- Manages the release schedule of VotaFun.
- Resolve any release-related issues and conflicts.

2.3. Team Communication

VotaFun communication channels include the following:

- Weekly face-to-face or online meetings are held every Thursday.
- Group announcements and updates are communicated via a dedicated Discord channel.
- Weekly updates are sent to the Discord channel every Monday.

3. Process Definition

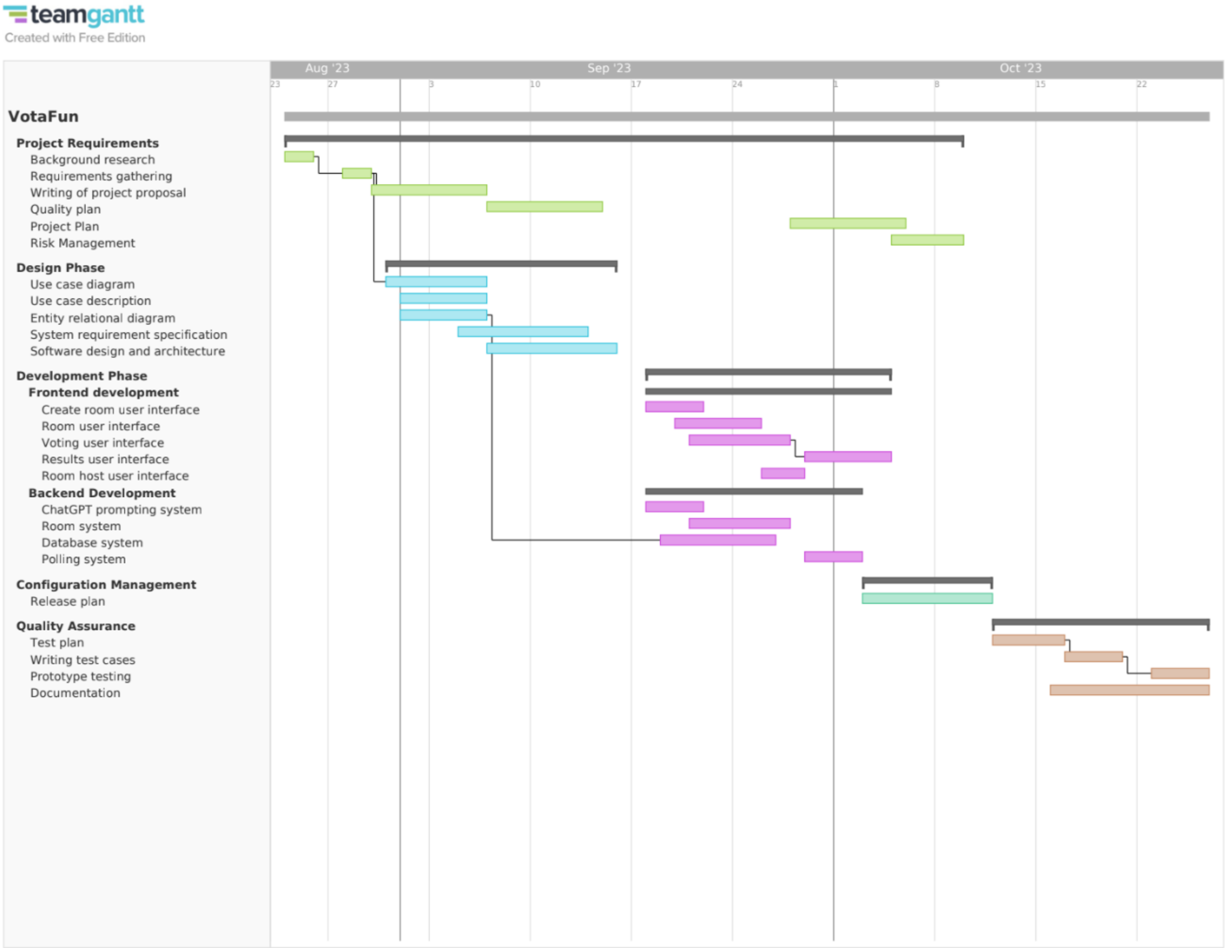
3.1. Lifecycle Model

The development team will use a hybrid of waterfall and agile methodology in this project. During the design phase, we will use the waterfall development methodology to ensure that all essential requirements and specifications are captured before any implementation begins.

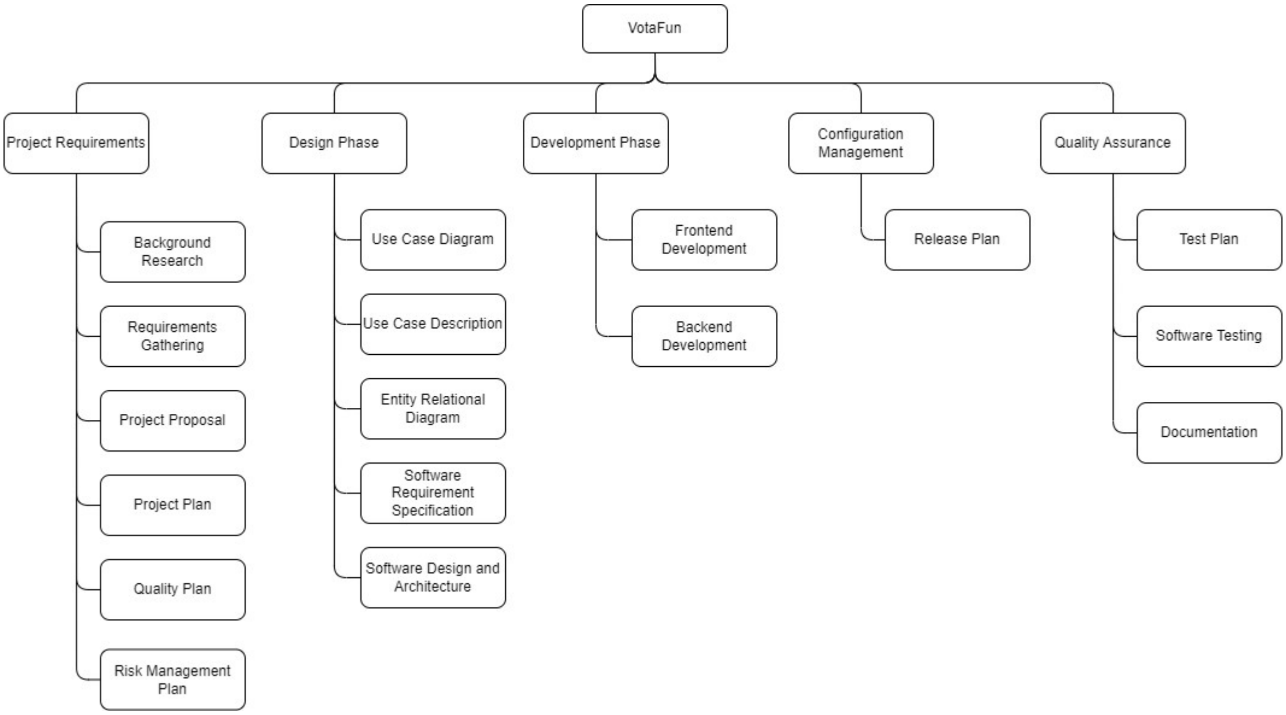
In the implementation phase, we will use the agile development methodology as it will give us the flexibility to translate VotaFun from design into code. The development team will continually gather feedback from the client and incorporate any feedback into VotaFun while building VotaFun incrementally. The team will also constantly update the requirements specifications, design plan and other documentation to reflect any changes to VotaFun. The team is separated into the front-end and back-end teams, which will both have sprint cycles of one week.

4. Schedule

4.1. Activity Dependencies and Schedule



4.2. Work Breakdown Structure



4.3. Work Packages

VotaFun is broken down into these critical phases of the software development lifecycle. These include:

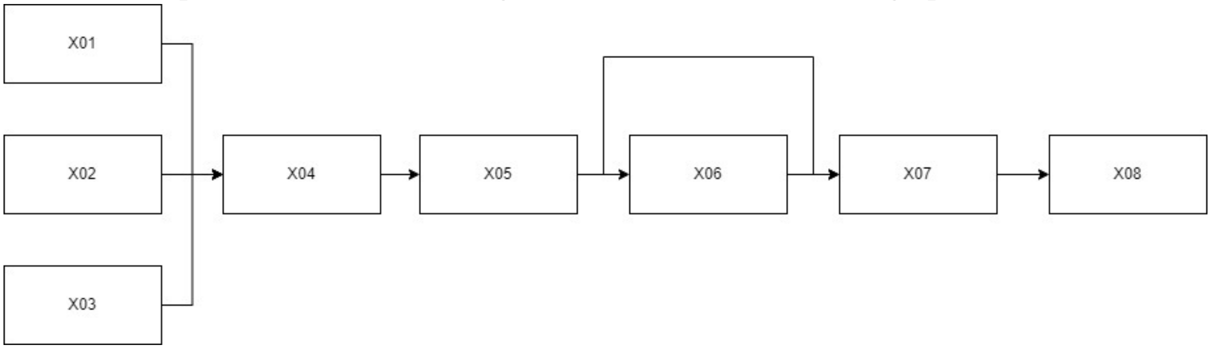
1. Project Proposal
2. Project Plan
3. Use Case Model
4. System Requirements Specification
5. Technical Architecture & User Interface
6. Data Modelling
7. Implementation & Unit Testing
8. Integration & Quality Assurance

4.4. Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work package #	Work Package Description	Duration (Days)	Dependencies
X01	Project Proposal	7 days	-
X02	Project Plan	7 days	-
X03	Use Case Model	7 days	-
X04	System Requirements Specification	7 days	X01, X02, X03
X05	Technical Architecture & User Interface	14 days	X04
X06	Data Modelling	7 days	X05
X07	Implementation and Unit Testing	21 days	X05, X06
X08	Integration and Quality assurance	14 days	X07

Below is a Dependencies Network Diagram to illustrate the above in graphical detail:



4.5. Work Package Details

Category	Description
Project	VotaFun
Work Package	X01 – Project Proposal (1 of 8)
Assigned To	**Ng Yue Jie Alphaeus**, Ryan Teo Cher Kean, Abdul Siddiq Bin Mohd Yussaini, Roy Lau Run-Xuan, Tran Trung Dung (Charles), Wang Xin Yan Lloyd
Effort	7PD
Start Date	Thu, 31/08/23
Purpose	Provide an overview of the project.
Inputs	None
Activities	Provides a detailed breakdown of the purpose, objectives, scope, and deliverables of the project. This work package involves gathering essential information, assessing feasibility, and outlining a comprehensive plan for project execution.
Outputs	A detailed document of the Project Proposal.

Category	Description
Project	VotaFun
Work Package	X02 – Project Plan (2 of 8)
Assigned To	**Ryan Teo Cher Kean**, Ng Yue Jie Alphaeus, Roy Lau Run-Xuan
Effort	7PD
Start Date	Thu, 31/08/23
Purpose	Provide a detailed timeline of the project deliverables.
Inputs	None
Activities	Provides a clear path forward for all stakeholders. It breaks down the

	project into manageable tasks, allocates resources, defines timelines, and establishes milestones, ensuring that the project progresses smoothly and successfully from initiation to completion.
Outputs	A detailed document of the Project Plan.

Category	Description
Project	VotaFun
Work Package	X03 – Use Case Model (3 of 8)
Assigned To	**Wang Xin Yan Lloyd**, Abdul Siddiq Bin Mohd Yussaini, Roy Lau Run-Xuan, Tran Trung Dung (Charles)
Effort	7PD
Start Date	Thu, 31/08/23
Purpose	Provide a structured and detailed representation of how VotaFun will be used by various entities.
Inputs	None
Activities	This work package involves the creation and documentation of a detailed Use Case Model, which serves as a key artefact for understanding, specifying, and visualising how VotaFun will function from the perspective of its users and other entities.
Outputs	A well-defined Use Case Model.

Category	Description
Project	VotaFun
Work Package	X04 – System Requirements Specification (4 of 8)
Assigned To	**Tran Trung Dung (Charles)**, Abdul Siddiq Bin Mohd Yussaini, Roy Lau Run-Xuan, Tran Trung Dung (Charles), Ryan Teo Cher Kean, Ng Yue Jie Alphaeus
Effort	7PD
Start Date	Thu, 7/09/23

Purpose	Provide documentation outlining the criteria for the system to be assessed.
Inputs	X01, X02, X03
Activities	This work package involves thorough analysis of user needs and expectations, team objectives, and technical constraints, resulting in a comprehensive document that acts as a blueprint for the entire project. The SRS will clearly and unambiguously describe the system's features, interfaces, data structures, and performance criteria.
Outputs	A detailed document of the System Requirements Specification.

Category	Description
Project	VotaFun
Work Package	X05 – Technical Architecture and User Interface (5 of 8)
Assigned To	**Roy Lau Run-Xuan,** Wang Xin Yan Lloyd, Abdul Siddiq Bin Mohd Yussaini, Tran Trung Dung (Charles), Ng Yue Jie Alphaeus, Ryan Teo Cher Kean
Effort	14PD
Start Date	Thu, 14/09/23
Purpose	Provide a high level overview of system components, system architecture and the design of the user interface.
Inputs	X04
Activities	This work package involves designing the software architecture, deciding different components needed and analysing the relationships between components. This work package will also identify the language and frameworks needed to build the project. The user interface will also be designed to show how users will interact with the application.
Outputs	A detailed diagram of the System Architecture that shows the design of the components and user interface.

Category	Description
Project	VotaFun
Work Package	X06 – Data Modelling (6 of 8)

Assigned To	**Wang Xin Yan Lloyd,** Ng Yue Jie Alphaeus, Abdul Siddiq Bin Mohd Yussaini, Ryan Teo Cher Kean, Roy Lau Run-Xuan, Tran Trung Dung (Charles)
Effort	7PD
Start Date	Thu, 28/09/23
Purpose	Design and come up with the entity relationship (ER) diagram which will be used to build the database.
Inputs	X05
Activities	This work package involves analysing how entities in VotaFun relate to each other, and also how data flows between these entities. The end result is an ER diagram that will clearly show the different entities, their attributes as well as how different entities relates to each other.
Outputs	A detailed ER diagram that models the database structure.

Category	Description
Project	VotaFun
Work Package	X07 – Implementation and Unit Testing (7 of 8)
Assigned To	**Abdul Siddiq Bin Mohd Yussaini, Ng Yue Jie Alphaeus, Ryan Teo Cher Kean, Wang Xin Yan Lloyd, Roy Lau Run-Xuan, Tran Trung Dung (Charles)**
Effort	21PD
Start Date	Thu, 5/10/23
Purpose	Development of the system according to the requirements specification, as well as testing individual components.
Inputs	X05, X06
Activities	Software engineers and developers are split into front-end and back-end teams. They will write, test, and integrate code components, adhering to established coding standards, architectural guidelines, and best practices. The correctness and functionality of individual components will be assessed to identify and rectify any defects, ensuring that the software is reliable and of good quality.

Outputs	Prototype of system

Category	Description
Project	VotaFun
Work Package	X08 – Integration & Quality Assurance (8 of 8)
Assigned To	**Abdul Siddiq Bin Mohd Yussaini,** Ng Yue Jie Alphaeus, Ryan Teo Cher Kean, Wang Xin Yan Lloyd, Roy Lau Run-Xuan, Tran Trung Dung (Charles)
Effort	14PD
Start Date	Thu, 26/10/23
Purpose	Ensure that the integrated VotaFun application adheres to the quality requirements as detailed in the previous documentation.
Inputs	X07
Activities	The work package encompasses verifying the interactions between software components, while maintaining and improving the overall quality of the software. Metrics are used to develop strategies for improving the software process, improving the quality of the end product.
Outputs	A detailed document of the Quality Assurance Report

5. Project Estimates

5.1. Code Size Estimation using Function Points

We calculated the unadjusted function point based on the complexity of functions provided by VotaFun. Code size is then estimated based on the adjusted function point.

5.1.1. Unadjusted Function Points (FP)

VotaFun supports the following proposed functions:

User Interface:

- Create a room, setting the room location and activity
- Display room location and activity
- Joining a room by entering the room code/ link
- Kick a user
- Vote for an option
- Display questions and options
- Display the final activity
- Viewing room information (Number of people, Current host)

Socket component (Python):

- Modify room status and update the database
- Add or kick a user in a room, and update the database
- Update the votes in the database when a user votes for an option
- Alert users if they are kicked
- Alert users if the room data changes

ChatGPT component (Python):

- Generates question and options, or activities for users
- Stores generated questions and options or activities

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Element	**Complexity**	**Detail**
Inputs	Medium	Vote for an option
	Low	Kick a user
	Medium	Joining a room by entering the room code/ link
	Medium	Create a room, setting the room location and activity

Outputs	High	Generates question and options, or activities for users
	Medium	Display questions and options
	Low	Display the final activity
	Medium	Alert users if they are kicked
	Medium	Alert users if the room data changes
	Medium	Display room location and activity
Inquiries	Medium	View room information (Number of people, Current host)
Logical Files	Medium	Add or kick a user in a room, and update the database
	Low	Modify room status and update the database
	Medium	Stores generated questions and options or activities
	Medium	Update the votes in the database when a user votes for an option
Interfaces	-	-

Calculation of Unadjusted Function Points:

Characteristic	**Low**	**Medium**	**High**
Inputs	1 x 3 = 3	3 x 4 = 12	0 x 6 = 0
Outputs	1 x 4 = 4	4 x 5 = 20	1 x 7 = 7
Inquiries	0 x 3 = 0	1 x 4 = 4	0 x 6 = 0
Logical files	1 x 7 = 7	3 x 10 = 30	0 x 15 = 0
Interfaces	0 x 5 = 0	0 x 7 = 0	0 x 10 = 0
Unadjusted Function Points (FP)	14	66	7
Total = L + M + H	14 + 66 + 7 = 87		

5.1.2. Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	5	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	4	Distributed processing and data transfer are online and in both directions.
Performance	5	Response time or throughput is critical for users to have a smooth experience.
Heavily used	4	Some security or timing considerations are included.
Transaction rate	5	Daily peak transaction period is anticipated.
On-line data entry	3	More than 30% of transactions are interactive data entry
End-user efficiency	5	Six of the efficiency designs are included
On-line data update	4	Distributed processing and data transfer are on-line and in both directions
Complex processing	2	Any two of the complex components
Reusability	2	Less than 10% of the application code developed is intended for use in more than one application.
Installation Ease	1	No special considerations were stated by the user, but special setup is required for installation.
Operational Ease	1	Start-up, back-up, and recovery processes were provided, but human intervention is required.
Multiple sites	2	The needs of more than one installation site were considered in the design, and the application is designed to operate only under similar hardware and/or software environments.
Facilitate change	0	None of the flexible query and report facility is needed.
Total score	43	

Influence Multiplier = Total Score × 0.01 + 0.65 = 43 × 0.01 + 0.65 = 1.08
Adjusted FP = Unadjusted FP × Influence Multiplier = 87 × 1.08 = 93.96

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3. Lines of Code (LOC)

Based on Quantitative Software Management, each FP requires 53 LOC when the application is implemented using Javascript, and 24 LOC when the application is implemented using Python[1].

LOC = 35.64 FP × 53 LOC/FP + 58.32 FP × 24 LOC/FP = 3288.6 LOC

5.2. Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (3288.6 LOC) / (39 LOC/PD)[2] = 85 PD (rounded up to the nearest integer)
- Duration = 3 × $(\text{Effort})^{1/3}$ = 3 × $(85)^{1/3}$ = 14 Days (rounded up to the nearest integer)
- Initial schedule = 14 Days / 5 days a week = 3 Weeks (rounded up to the nearest integer)
- Team size = 85 PD / 14 D = 6.07 P = 7 Persons (rounded up to the nearest integer)
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 85 PD × 8 hours = 680 PH

[1] Our team used Perl average LOC/FP as a substitute for Python LOC/FP as both languages are similar. Quantitative Software Management. n.d. Function Point Languages Table. https://www.qsm.com/resources/function-point-languages-table. Online; accessed 4 October 2023

[2] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

Based on the estimates, VotaFun will take 3 weeks to complete if there are 7 members in the team working 8 hours a day, 5 days a week.

5.2.1. Distribution of Effort

1990's Industry Data	**Work Package**	**Distribution**	**Estimates**
Preliminary Design 18 %	Project Plan	6%	40.8
	Project Proposal	6%	40.8
	Use Case Model	6%	40.8
Detailed Design 25 %	System Requirements Specification	8%	54.4
	Technical Architecture & User Interface	11%	74.8
	Data Modeling	6%	40.8
Code & Unit Testing 26 %	Implementation & Unit Testing	21%	142.8
	Online Documentation	5%	34
Integration & Test 31 %	Integration & Quality Assurance	31%	210.8
	Extrapolated total effort		680
	2% for project management		13.6
	3% for contingency		20.4
	Total effort		714

5.3. Cost Estimates

Item	**Supplier**	**Quantity**	**Unit Price (SGD)**	**Cost per month (SGD)**	**Total (3 months)**

Project Manager	-	1	-	$7,000.00	**$21,000.00**
Software Developers	-	3	-	$5,500.00	**$49,500.00**
QA/Release Engineers	-	3	-	$5,000,00	**$45,000.00**
Laptops	Dell	6	$1,090.00	-	**$5,540.00**
Printer	Fuji Xerox	1	$450.00	-	**$450.00**
Technology Licence	Visual Paradigm	1	$99.00	-	**$99.00**
Cloud Services	Heroku	1	-	$300.00	**$900.00**
Office rental	NTU	1	-	$5,000.00	**$15,000.00**
				TOTAL	**$137,489.00**

6. Product Checklist

Project Deliverable	**Deadline**
Project proposal	7 September 2023
Use Case Model	7 September 2023
System requirement specifications	21 September 2023
Quality assurance plans	21 September 2023
Project plan	12 October 2023
Risk management	12 October 2023
Prototype demo	12 October 2023
Design report on software maintainability	26 October 2023
Configuration management plan	26 October 2023
Change management plan	26 October 2023
Release plan	26 October 2023
Presentation slides	26 October 2023
Test plan	26 October 2023

Test cases and requirements test coverage report	26 October 2023

7. Best Practice Checklist

Practice	
Document what we do; all documentation must be in a standardised format.	
Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification.	
Keep it simple. Complexity management is one of the major challenges. Strive to: ● Minimise interfaces between modules, procedures and data. ● Minimise interfaces between people, otherwise exponential communication cost ● Avoid fancy product functions, design as long as the functionality meets the customer requirements	
Require Visibility. We must see what we build otherwise we can measure the progress and take management action. This includes: the manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness.	
Plan for continuous change. We must: ● All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes ● New revisions should be approved before being made and checked for quality and compliance after being made ● Use a configuration management system and make processes ● Required maintenance	
Don’t underestimate the project. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.	
Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members	
Software testing will use both black box and white box testing. It will involve unit, functional, integrating and acceptance testing.	

8. Risk Management

Besides the general risk management, the following risks have been identified for VotaFun:

1. More changes to requirements than anticipated

Impact Severity: High
Probability: 20%
Impacts: With a focus on accommodating frequent changes, there is a risk that the team may overlook or inadequately address important requirements. This may require minor changes to the requirements specification or a redesign of the application.
Risk Reduction: The development team will be as detailed as possible when formulating requirements. The team will also notify the client of the consequences of any changes.

2. Launch of new large language models (LLM)

Impact Severity: Medium
Probability: 5%
Impacts: The release of new LLM might render VotaFun obsolete as it could potentially be better than ChatGPT.
Risk Reduction: VotaFun has to accept the risk of new LLM being developed and released. The development team will ensure that the system is designed to be as maintainable as possible to incorporate any new technologies that will appear.

3. Underestimation of project size

Impact Severity: High
Probability: 25%
Impacts: Possible schedule delays or quality compromises. There is also an increased chance of other risks coming true as project team members rush to complete deliverables, which may compromise the quality of VotaFun.
Risk Reduction: The team must have a better gauge and estimate the project size accurately at all stages of the project.

4. Project running behind schedule

Impact Severity: High
Probability: 10%
Impacts: The team cannot deliver a prototype to the client on time for testing and usage, which will incur even more cost.
Risk Reduction: The project manager must ensure that the project schedule is followed closely and that the burndown chart does not diverge.

5. Disagreements within the project team

Impact Severity: Medium
Probability: 50%
Impacts: This may lead to strained relations within the project team, which can affect team progress.
Risk Reduction: The project manager must ensure that disagreements between team members are mediated. Team members must surface any disagreements to the project manager quickly.

6. Key personnel leaving the team

Impact Severity: Medium
Probability: 10%
Impacts: The team will lose any specialised skills and knowledge that the team member has. The remaining team members will also have an increased workload.
Risk Reduction: The team will encourage cross-training to ensure that another member can step in when necessary. The team will also give incentives and benefits to encourage members to stay.

7. Underestimation of budget needed for the project

Impact Severity: High
Probability: 15%
Impacts: This may result in possible delays as the team has to request additional funding from the client. It also reduces the trust in our team to manage projects.
Risk Reduction: The team must come up with an accurate estimation of the budget needed for the project. The team will regularly track the budget and update costs if there is any anticipated change.

8. Dependency on 3rd party APIs

Impact Severity: High
Probability: 20%
Impacts: Users will experience disruptions to the service if ChatGPT becomes unavailable due to maintenance or other reasons.
Risk Reduction: The team has to accept this risk as the core functionality of VotaFun requires ChatGPT. The team will also research incorporating other LLM so that VotaFun remains usable if any one of them is unavailable.

9. Market changes resulting in less uptake

Impact Severity: High
Probability: 10%
Impacts: VotaFun may be less successful than anticipated and a lower number of users will use VotaFun.
Risk Reduction: The team will conduct regular market research to keep ahead of users' preferences and current trends to ensure that VotaFun will be successful.

9. Quality Assurance

The project will achieve quality assurance by following the standard set. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Test Plan.

In addition, VotaFun shall make use of the following testing methodologies:

- Unit testing – Each individual component will be tested individually.
- Integration testing – Components are tested together to ensure that they can work well.
- System testing – All components will be linked together, and testing will be done to ensure that all functional and quality requirements are met.
- User acceptance test – A beta application will be set up and the client will be invited for testing.

Broad and realistic test cases will be used during each of the testing phases. Additionally, the client will be invited to perform a user acceptance test to ensure that it has met all functionalities outlined in the project plan, and requirements specifications. Test cases, as well as testing results, will be given as well.

10. Monitoring & Control

This section outlines the processes, activities, and tools that will be used to track, assess, and manage the project's progress, performance, risks, and changes throughout its lifecycle. There are several key components that are listed below:

1. **Performance Metrics and KPIs:** Define the specific performance metrics and key performance indicators (KPIs) that will be used to measure project progress and success. These metrics include schedule adherence, budget variance, quality indicators, and more.

2. **Project Tracking:** Specify how project progress will be tracked, including the frequency and methods for collecting data and monitoring project activities. This involves regular status meetings as specified in earlier sections, progress reports, and project management software such as Heroku.

3. **Risk Management:** Describe the processes for identifying, assessing, and managing project risks. This includes defining risk mitigation and contingency plans and assigning responsibilities for risk management activities.

4. **Quality Assurance and Control:** Detail how quality assurance and quality control activities will be performed to ensure that project deliverables meet the required quality standards. This will include inspections, reviews, and testing processes.