VOTAFUN

RELEASE PLAN

Version 1.2
01/11/2023

APPROVALS

Submitting Organization’s Approving Authority:

alphaeus.	Ng Yue Jie Alphaeus	25/10/2023	9000 0000
Signature	Printed Name	Date	Phone Number

Project Manager

Position Title

REVISION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ng Yue Jie Alphaeus	12/10/2023	Ng Yue Jie Alphaeus	12/10/2023	Initial release plan draft
1.1	Wang Xin Yan Lloyd	18/10/2023	Ng Yue Jie Alphaeus	25/10/2023	Introduction added Referenced documents added Overview added Assumptions, constraints and risks added Release approach added Acronyms added
1.2	Roy Lau Run-Xuan	31/10/2023	Ng Yue Jie Alphaeus	01/11/2023	Technical risk and management risk added

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This release plan is a detailed specification of the features of Votafun, the upcoming release versions and the release dates. This release plan aims to aid the team with understanding the team’s progress, goals, and priorities. Having this release plan helps the team identify the criteria and targets for each release. Continuous integration is practised along the project timeline as developers continually add features.

The intended audience for the Release Plan is the entire project team. The development team will build the release version, the QA team is responsible for the quality management of each release, and the project manager overseas and approve each release.

2 REFERENCED DOCUMENTS

Table 1: Referenced Documents

Document Name	Issuance Date
Project Proposal	7/9/2023
Software Requirement Specification	21/9/2023
Risk Management Plan	12/10/2023
Change Management Plan	26/10/2023

3 OVERVIEW

Votafun is a multiplayer Kahoot-style web application. Its target audience is groups of people looking to plan activities such as outdoor excursions, lunch plans or even movies to watch.

The following are objectives that VotaFun should achieve:

1. **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.

2. **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.

3. **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.

4 ASSUMPTIONS, CONSTRAINTS, RISKS

4.1 Assumptions

Schedule:

1. The development and release timeline should not be altered considerably.

2. The team members are willing to commit the required amount of time for development.

3. All tasks are completed as per the schedule.

Budget and Financing:

1. There are no sudden budget cuts or hidden costs.

2. The budget estimation and resource allocation for Votafun is precise.

Development:

1. The developers are skilled and able to meet expectations.

2. The developers and customers can collaborate on the development activities to get proper outcomes.

Technology:

1. The main framework of the system is to remain unchanged after release.

2. Cloud services and ChatGPT are available throughout the whole development process.

4.2 Constraints

Manpower:

1. The team size is relatively small. Careful work assignments and task planning must be carried out. The Project Manager must exercise a structured and flexible manpower allocation to tasks to easily move developers around the frontend and backend teams to ensure the project tasks can be completed on time.

Time:

1. The entire project from the planning to the release stage is to be completed in a short time frame, between August to November 2023. To ensure no delays, project planning must be cautious and correctly estimated.

4.3 Risks

Table 2.1: Technical Risks

Risk ID	RISK 14
Risk Description	Requirements inflation where clients ask to add new features not identified during the planning phase. Depending on the size of these features, it may threaten the project timeline and risk the project team missing the dateline.
Risk Response	Mitigate
Probability of Occurrence	15%
Impact Level	High
Contingencies	The project manager has to communicate with the client and let them understand the impact and consequence of adding new features. The project manager must capture all requirements and clarify discrepancies before any implementation starts.

Risk ID	RISK 15
Risk Description	Different developers have different experience levels or coding styles that may lead to poor code quality. Adding new features or maintaining the software becomes more difficult. Poor code quality may introduce more bugs as well.
Risk Response	Avoidance
Probability of Occurrence	20%
Impact Level	Medium
Contingencies	The development team will practice pair programming, where experienced and inexperienced members are paired. The development team also uses pre-commit hooks to standardise the code format. The team lead will also perform a code review before merging code.

Risk ID	RISK 16
Risk Description	The lack of client engagement may result in the developer team adding redundant features. The developer team will waste time developing redundant features.
Risk Response	Mitigate

Probability of Occurrence	10%
Impact Level	Medium
Contingencies	The project manager will constantly engage the client to ensure that the team don’t waste time on redundant features not needed by the client.

Table 2.1: Management Risks

Risk ID	RISK 17
Risk Description	The project team may lack the urgency to work during the earlier stages of the project due to the assumption of a long project timeline.
Risk Response	Mitigate
Probability of Occurrence	25%
Impact Level	Medium
Contingencies	All team members must submit their project deliverables by the end of each sprint cycle. The project manager will communicate goals and deliverables to deliver before at the end of a sprint. The project manager will talk to team members who are not meeting expectations.

Risk ID	RISK 18
Risk Description	The project ran behind schedule due to poor timeline estimates or unforeseen circumstances that prevented the team from meeting the release cycle.
Risk Response	Mitigate
Probability of Occurrence	30%
Impact Level	Medium
Contingencies	The project manager will allocate extra time during the timeline estimation to account for unforeseen events. The project manager expects constant communication from developers to ensure the team is on track for any release.

Risk ID	RISK 19
Risk Description	Key personnel may leave the team. The shortage of experienced team members may result in the project team

	not meeting the release cycle. The remaining team members will face an increased workload, which may decrease morale.
Risk Response	Accept
Probability of Occurrence	20%
Impact Level	Medium
Contingencies	Incentives and benefits are given to encourage team members to stay. The project manager will also immediately look for new members to join the project team. Pair programming practice may also help mitigate the risk of key personnel leaving.

5 RELEASE APPROACH

5.1 Rationale

VotaFun will be built and developed by the team in incremental phases with multiple releases as the project team has chosen the agile software development methodology to develop VotaFun. Each release cycle will build upon the previous version. Features may be added or removed, and bugs fixed.

Version 1.0.0 will serve as a baseline version for VotaFun. Subsequent changes to VotaFun will require the review of the CCB. If changes are approved, the developer team will work to release changes in the next version of VotaFun.

5.2 Release Strategy

The project team will release VotaFun in different release cycles using a phased rollout approach, with each cycle aiming to improve on the previous versions. Version 1.0.0 of VotaFun will be the baseline version that implements all functionalities written in the SRS. Subsequent releases after Version 1.0.0 will be to maintain VotaFun, bug fixes, or add or remove features.

A major release happens when the project team performs a huge overhaul of VotaFun with multiple features added or removed. Multiple components are affected in a major release. A minor release happens when there are minor improvements to the functionality or performance of the system. Some bug fixes may be set for a minor release, especially if it affects the functionality of VotaFun. Finally, a patch is for small changes that do not affect functionality or to fix bugs.

5.3 Release Content

Following the release strategy, the following table shows the different release versions scheduled for VotaFun. The team will implement and test all release versions to ensure that they fulfil the aims of the release.

Table 3.1: Initial Release

Release Version 1.0.0

Functionality	Description
Front-end Interactive Webpage: - Create/ join room - Lobby page - Session page	Main functions implemented as defined in initial SRS
Socket integration	Fully implemented as defined in initial SRS
Redis database	Fully implemented as defined in initial SRS
ChatGPT prompt engineering	Fully implemented as defined in initial SRS

Table 3.2: Minor Release

Release Version 1.1.0	
Functionality	Description
Front-end Interactive Webpage: - Lobby page - Session page	Bug fixes and minor improvements. ● Leave and close room buttons. ● Users can set the max capacity of rooms.
ChatGPT prompt engineering	Minor improvements ● Improvements of prompt engineering

Table 3.3: Major Release

Release Version 2.0.0	
Functionality	Description
Front-end Interactive Webpage: - Summary Page - Regenerate question and options	Introduction of new major features or system overhaul ● Summary page which summarises the votes for all questions. ● Users can regenerate questions and options if they are not satisfied with the questions and options shown.
Socket integration	Introduction of new major features or system overhaul
Redis database	Introduction of new major features or system overhaul ● Improved redundancy and multiple backups ● Addition of durable storage (ie traditional DB)
New large language models (LLMs)	Introduction of new major features or system overhaul ● New LLMs for users to use instead of only ChatGPT.

5.4 Release Schedule

The project team will deploy the first release on 26 October 2023, with all functionalities in the SRS implemented. The team is scheduling to release the minor release (version 1.1.0) 3 months later and the major release (version 2.0.0) 5 months after the minor release. These release estimates will include the time for the change management process, rewriting of documentation, implementation, testing and release process.

5.5 Release Impacts

For the minor release (version 1.1.0), there will be minor impacts to VotaFun, namely the frontend component and the ChatGPT component. Whereas for the major release (version 2.1.0), more components are affected with the backend components (redis and ChatGPT) scheduled to have more changes. The project team will follow the change management process before working on the new releases. The project team will rewrite documentations like the system architecture, SRS, and use case model for both releases. The project and release manager shares the responsibility to mitigate any possible risks during each release.

The project team will push new releases between 12:00 a.m. and 06:00 a.m. (GMT +8), thus ensuring that users are not affected by new releases of VotaFun. For critical releases due to security vulnerabilities, the team will deploy VotaFun as quickly as possible.

5.6 Release Notification

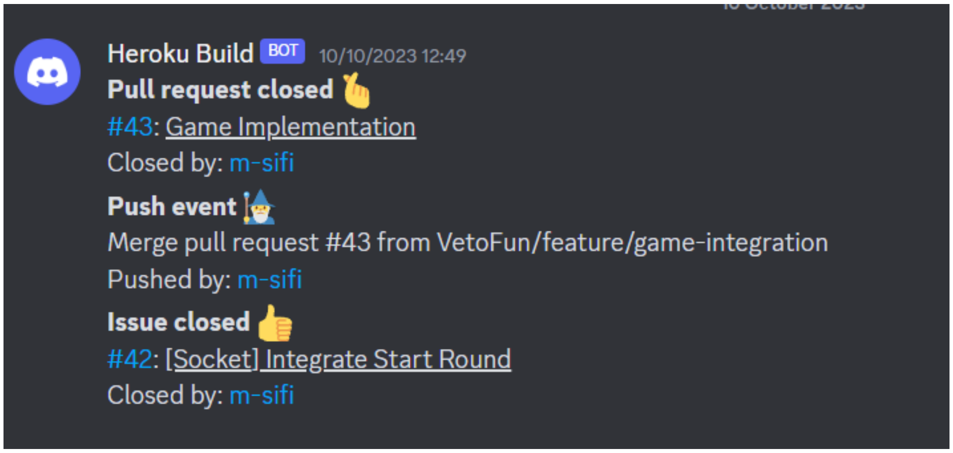
Before the deployment of a release version, the project team will notify different stakeholders of the changes. The following table shows the stakeholders notified, notification methods, information included and when the project team notifies stakeholders.

Table 4: Release Notification

Stakeholders	Notification method	Information included	Timeframes for receipt of information
Client	● Emails ● Virtual meetings	● Change log	One week before a new deployment
VotaFun users	● System Notifications	● Change log	After the team deploys a new release.
Team Members	● Meetings ● Emails ● GitHub discord bot	● Changes to make ● Known issues / bugs to fix ● Deadlines of release implementation	After the CCB has confirmed the change.

Furthermore, we have set up a Github Discord Bot integration that will notify us of all important events related to the development and release of the application.

Figure 1: Github Discord Bot



10 October 2023

Heroku Build BOT 10/10/2023 12:49

Pull request closed 🫰

#43: Game Implementation

Closed by: m-sifi

Push event 🧙

Merge pull request #43 from VetoFun/feature/game-integration

Pushed by: m-sifi

Issue closed 👍

#42: [Socket] Integrate Start Round

Closed by: m-sifi

The figure above is an example of some messages sent from the bot to our Discord channel. Having the bot keeps the team updated on issues and pull requests that require our attention. The discord bot notifies all push events to the main (production) branch in the project team's discord channel, which allows the team to be clear on the development state.

6 ACRONYMS

Table 5: Acronyms

Acronym	Phrase
CCB	Change Control Board
SRS	System Requirement Specification
SDLC	Software Development Lifecycle
QA	Quality Assurance