

School of Computer Science and Engineering

CZ3002 Advanced Software Engineering

**Release Plan**

Project Name: HangOut

****

Group Name: Mac & Cheese

Lab Group: TDDP1

| **Group Members** |
| --- |
| Jethro Phuah An Ping (Team Leader) |
| Lam Zhi Fah (Deputy Team Leader) |
| Alicia Chua Jieying |
| Ernest Ang Cheng Han |
| Ong Jing Heng Shaun |
| Ong Sim Hao |
| Shannon Tan Xinyi |

# Revision History

| **Revision Number** | **Date** | **Primary Author(s)** | **Comments** |
| --- | --- | --- | --- |
| 1.0 | Oct 1st, 2021 | Jethro Phuah, Alicia Chua, Shannon Tan, Lam Zhi Fah, Shaun Ong, Ong Sim Hao, Ernest Ang | First version |
|  |  |  |  |

**TABLE OF CONTENTS**

[**1 Introduction**](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.j6b62kz989k2) **2**

[**2**](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.irwusy3ku6fp) **Referenced Documents 3**

[**3 Project Overview**](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.yy0a6mdixb6o) **3**

[**4**](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.c5xfros1x9r2) **Assumptions, Constraints, Risks 5**

[4.1 A](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.hxwfbdfa9pn)ssumptions 5

[4.2](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.vwskdyall74r) Constraints 5

[4.3](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.s67ui14coru) Risks 6

[**5**](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.ejn19uz0w0qi) **Release Approach 7**

5[.1](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.hxwfbdfa9pn) Rationale 7

5[.2](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.vwskdyall74r) Release Strategy 8

5[.2.1](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.s67ui14coru) Release Content 8

5[.2.2](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.s67ui14coru) Release Schedule 9

5[.2.3](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.s67ui14coru) Release Impact 10

5[.2.4](https://docs.google.com/document/d/1fWgg5RfhWLweQ_VIAVYzdNOXnbrfsRJX/edit#heading=h.s67ui14coru) Release Notification 11

**1. INTRODUCTION**

This Release Plan documents the planning and deployment of new features to our HangOut application system through incremental release plans. Each release plan has a unique id with the following format: HangOut X.Y.Z’, where X refers to major releases, Y refers to minor releases, and Z refers to revisions. Major releases refer to huge HangOut (usually of architectural scale) to the HangOut application, minor releases refer to addition of new features (e.g. upgrades to our meet-up features), and revisions refer to hot-fixes and patches. For each release plan, we aim to outline the features to be developed and tested before deploying them and writing notes for maintenance, providing useful information to make decisions on how much functionality is required in the upcoming release as well as how long and how much effort would be required by the team to meet important deadlines and project milestones. This release plan is for all members within the project team only directly involved with the management, design and development of HangOut .

A Non-Disclosure Agreement is automatically entered into by and between all parties reading this document, and with Team MacAndCheese for the purpose of preventing the unauthorized disclosure of Confidential Information. The parties agree to enter into a confidential relationship with respect to the disclosure of certain proprietary and confidential information. If you are in possession of this documentation and are not authorised to view this documentation, kindly and promptly dispose of it while adhering to this non-disclosure clause to prevent any form of legal action to be pursued by Team MacAndCheese’s Legal Team.

**2. REFERENCED DOCUMENTS**

This release plan makes references to the below mentioned documentations and diagrams. Do refer and familiarise yourself with them in order to better and fully obtain a high level overview of the entire HangOut application project.

**Table 1: Referenced Documents**

| **Document Name** | **Document Number** | **Issuance Date** |
| --- | --- | --- |
| Mac & Cheese Use Cases | Version 1.0 | 22 August 2021 |
| Mac & Cheese Project Proposal | Version 1.0 | 22 August 2021 |
| Mac & Cheese SRS | Version 1.0 | 12 September 2021 |
| Mac & Cheese Quality Plan | Version 1.0 | 12 September 2021 |
| Mac & Cheese Project Proposal | Version 1.0 | 23 September 2021 |
| Mac & Cheese Risk Management Plan | Version 1.0 | 23 September 2021 |

**3. OVERVIEW**

HangOut was conceptualised to promote a wholesome and tight-knitted community within Singapore, in order to bring back one of Singapore’s treasured yet dying heritages - the “Kampung” spirit, referring to the unity, solidarity and harmony displayed amongst residents way before the concrete Jungle that Singapore is today, during the 20th Century. In light of urbanization and industrialisation, Singaporeans have begun to shut themselves out and focus on career-centric goals as our economy develops and diversifies. Covid 19 has also done nothing to enable Singaporeans to step out of their inner circles to bond with their community given the many restrictions on social interactions and unpredictable changes to our Covid 19 alert levels. Through HangOut, we hope to bring back this dying heritage via encouraging safe interactions within one’s own community as well as to piques one’s interest of the various regions in Singapore by providing information on tucked-away hidden gems revolving around places of interests, food places, and events being held around a certain constituency. We hence aim to encourage interest, pride, and social interaction within one’s own community.

HangOut will consist of 2 interfaces: the User interface and the Administration interface. The user interface will be a mobile application for users to look up food options, places of interests and events occurring in any particular region of Singapore. Users will also be able to create and organise ‘meetup’s to encourage bonding and wholesome befriending amongst a community. Given that it is a mobile application, it is hence easily accessible by most Singaporeans according to a report by Statista.com which states that as of 2020, nearly 5.17 million Singaporean residents use a smartphone. The mobile application will be available on both iOS and Android platforms, the top 2 most popular used mobile operating systems in Singapore as reported on GlobalStats Counter. The administration interface will be a web application for administrators from the various regions (i.e. organised by Group Representation Constituencies), to create, read, update and delete (CRUD) food options, events, and places of interests around the vicinity of their respective region. All of these changes to their data will be updated in real-time over on the User interface.

We have decided to implement a 3 tier-architecture to design and develop this project due to the benefits of abstracting the entire application into 3 separate layers: Presentation, Business Layer, Database Layer. This will allow us to design the User interfaces without having to worry about the backend business logic interacting with the data models, useful in this case given that we have 2 different interfaces (user and administration). We will also be deploying both interfaces live into production as observed in Figure 1.1 displaying the deployment diagram and overall methodology we intend to use for the HangOut Project - building the mobile application into APK/IPA files while deploying the Administration interface on a suitable server, most likely via softwares offering Platforms-as-a-Service (Paas) such as Heroku and Netlify.

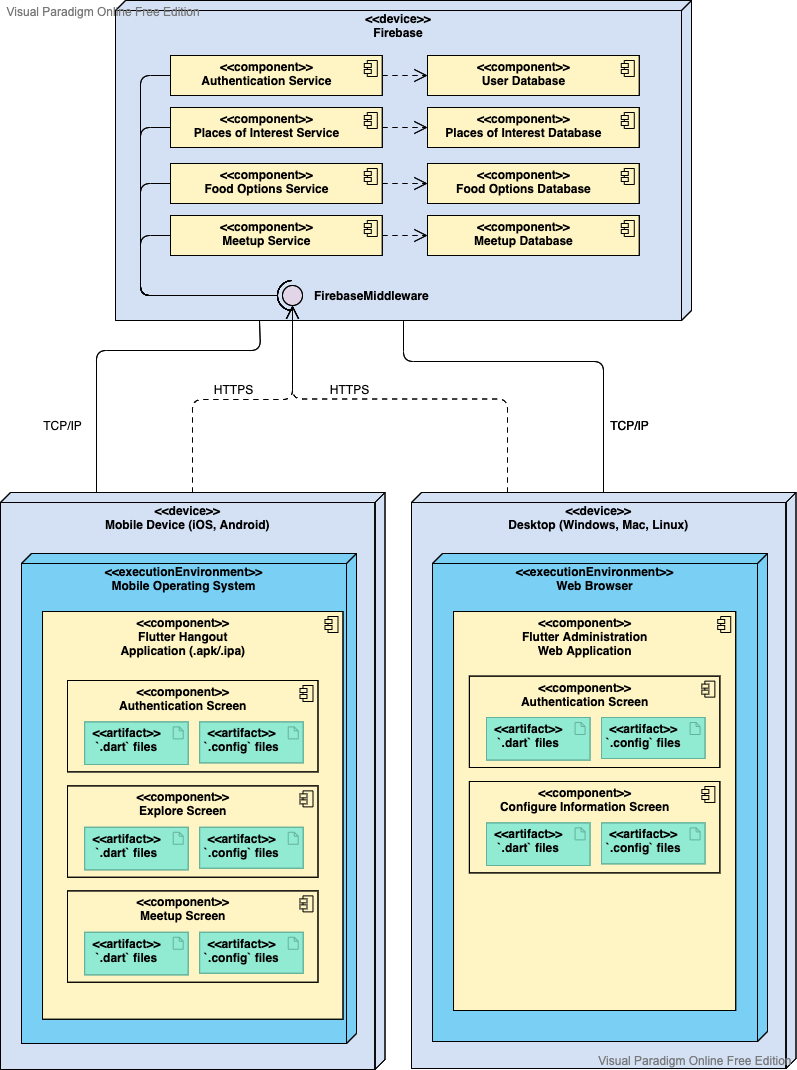


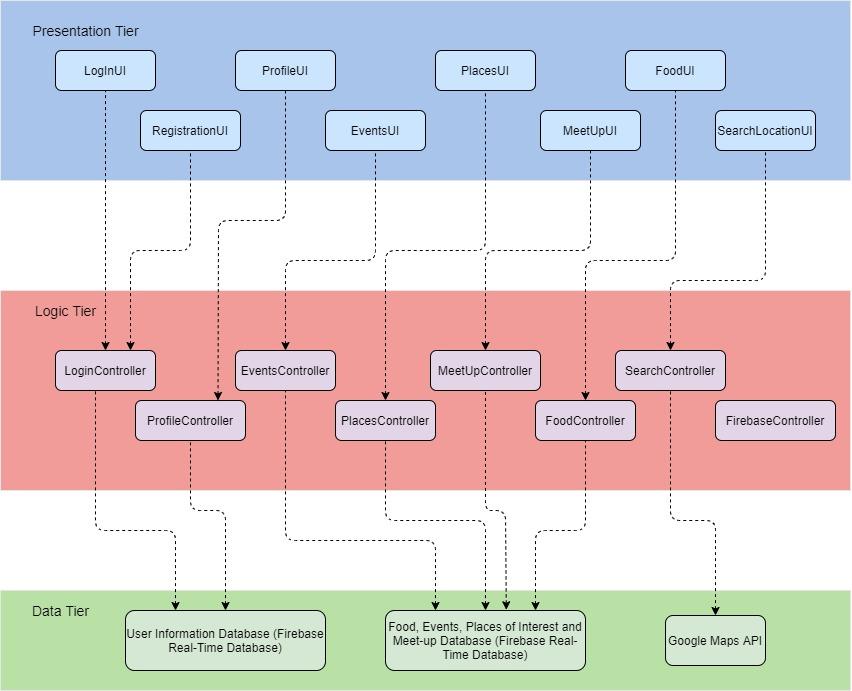
Fig 1.1 HangOut Deployment Diagram 

Fig 1.2 HangOut Multi-Tiered Architecture

**4. ASSUMPTIONS, CONSTRAINTS, RISKS**

**4.1. Assumptions**

For the current system, we assume that we will have the necessary development resources, mostly in terms of manpower strength given that we are mostly using free-tiered tooling and open-source softwares, to build the entire project. In light of injuries or offboarding of team members, we also assume that it would be easy to source and onboard new engineers that will be able to quickly pick up and complete tasks in a quick and agile fashion as well as understand the framework of choice (Flutter, Dart and Firebase) and the code base. We also assume that all our software dependencies will be available and resilient, without internal bugs and security flaws that the project team will otherwise have to backlog and find workarounds as well as alternatives so as to not impact the functional and non-functional requirements of HangOut specified in our Software Specification Requirement documentation. With the professionalism and experience of every group member, we can also assume that every individual will put their best effort to be punctual and actively participate in all meetings while avoiding groupthink, as well as hit project and engineering milestones to allow projects to see progress without creating bottlenecks.

**4.2. Constraints**

One of the technical constraints is the development of the Administration interface using Flutter instead of a conventional and more optimized web framework such as Bootstrap or React JavaScript. Flutter is originally designed as a platform to develop cross-platform mobile applications suitable for both Android and iOS operating systems, which is a great advantage when designing the mobile application. Flutter does provide support for developing web applications but is far weaker and slower in terms of optimization as well as the amount of resources available due to the lack of popularity in using Flutter to develop websites in the first place. However, given that we are using Firebase, a Backend-as-a-Service, we will most likely be using Dart to connect to Firebase. Hence, in using Flutter which is based on Dart, to develop our Administration interface, we will reduce the development time in translating the Dart-written connection to Firebase to another language (e.g. JavaScript if React was used instead of Flutter). This constraint is ultimately a trade-off with reducing additional and unnecessary development time for more backend-related work.

On the other hand, when it comes to evaluating more Business-related constraints, the short timeline coupled with the intense nature of our project requirements and milestones given the project team size and capabilities are the greatest restrictions that have the potential to delay and derail the release of HangOut. The project management team has to be aware of the limitations of the engineering team when planning sprints or performing backlog grooming in order to ensure that the milestones are still achievable albeit with much difficulties and we will be able to observe incremental progression towards the eventual completion and release of HangOut. The project management team also needs to familiarize themselves with the 6 common constraints recognized by the Project Management Institute (PMBOK) as the main determining factors in the successful delivery of a project: Scope, Quality, Schedule, Budget, Risk and Resources.

**4.3. Risks**

There are a plethora of risks associated with the release of HangOut. The first of which is the inaccuracies revolving around estimations made throughout the management of the entire project. Although the usage of approximation in some figures and statistics of the project is unavoidable, these inaccuracies create risk especially to end-users and other important stakeholders when expectations are promised but not eventually met (e.g. overestimating the team’s abilities and promising unrealistic release dates). An easy solution for this would be being level-headed and practical during planning each sprint, ensuring that each iteration is short with manageable and achievable tasks and goals, creating many opportunities for team members to have retrospective sessions on what is beneficial in moving the project while highlighting certain pain points slowing down the team.

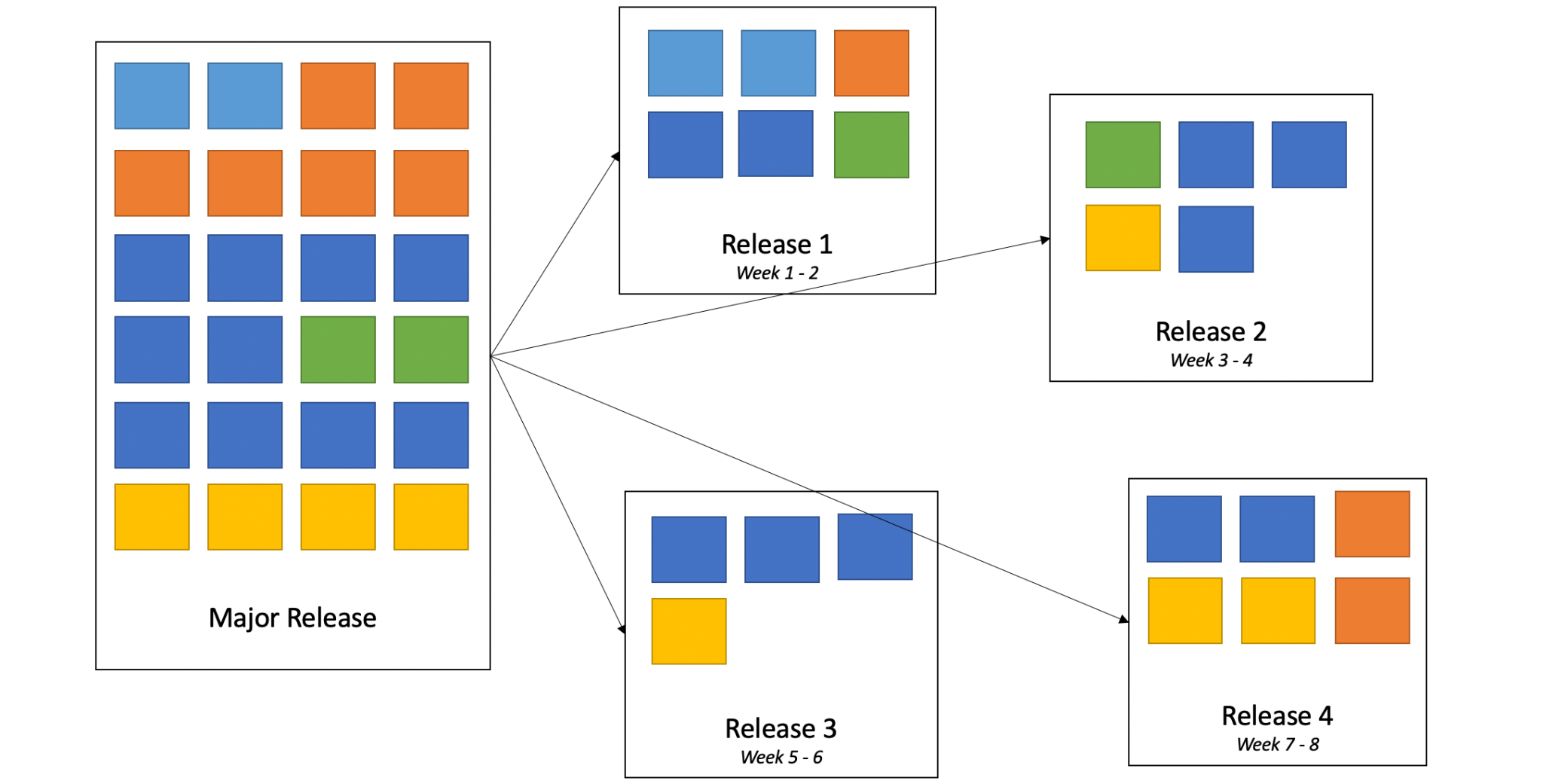
It is important to manage inaccurate estimation as this exacerbates the next risk, which is end-user engagement. This is important and must take place prior to the inception of the project given that we will need to ensure that the target audience of our product will actually see the value in the design of HangOut and will go on to eventually use and benefit from our product. From a practical and financial point of view, this definitely correlates with profitability and the strategy in having high levels of engagement with end-users is much more straightforward and implementable - listening to them. Arrange for user testing and conduct surveys, conduct ample amount of balanced research to ensure that there is indeed a great and dire need and anticipation for an application such as HangOut.

Finally, another source of concern is inadequacy in risk management. This occurs the moment any portion of the project’s identified risks are not properly dealt with by the relevant and involved parties. This is unfortunately common in many software engineering teams and project managers need to be proactive in not just identifying such insufficiencies but being deliberate in planning action steps in mitigating these risks and subsequently delegating responsibilities to executing these strategies. Some mitigation strategies include accounting for risk itself during project estimation, as well as using a Risk Register on our estimations and backlog. Additionally, project managers should perform risk analysis at the start and end of every single iteration meeting to make the mitigation strategies even more all-encompassing.

**5. RELEASE APPROACH**

**5.1. Rationale**

We will be using Agile release planning in order to organise and execute the delivery of HangOut. Instead of the traditional approach of releasing major version updates with usually huge numbers of features added to a product, Agile release planning batches the release of new features in smaller minor versions and patches after every few iterations in the Agile sprint cycle. The project managers plan and prepare for a major release (e.g. adding 5 new features to the meetup functionality on the user interface of HangOut) before then separating it into several portions which will be developed and completed in Agile sprints or iterations. The rationale behind this methodology as opposed to the conventional release of huge changes in major updates at one go is due to the increasing complexities revolving around server-side and client-side requirements. Along with the desire for flexibility by end-users and business stakeholders, breaking up major releases into small sprints simplifies the tasks needed to be completed by the product release date and gives room for stakeholders to pitch in changes or improvements to the features being developed in the current sprint.



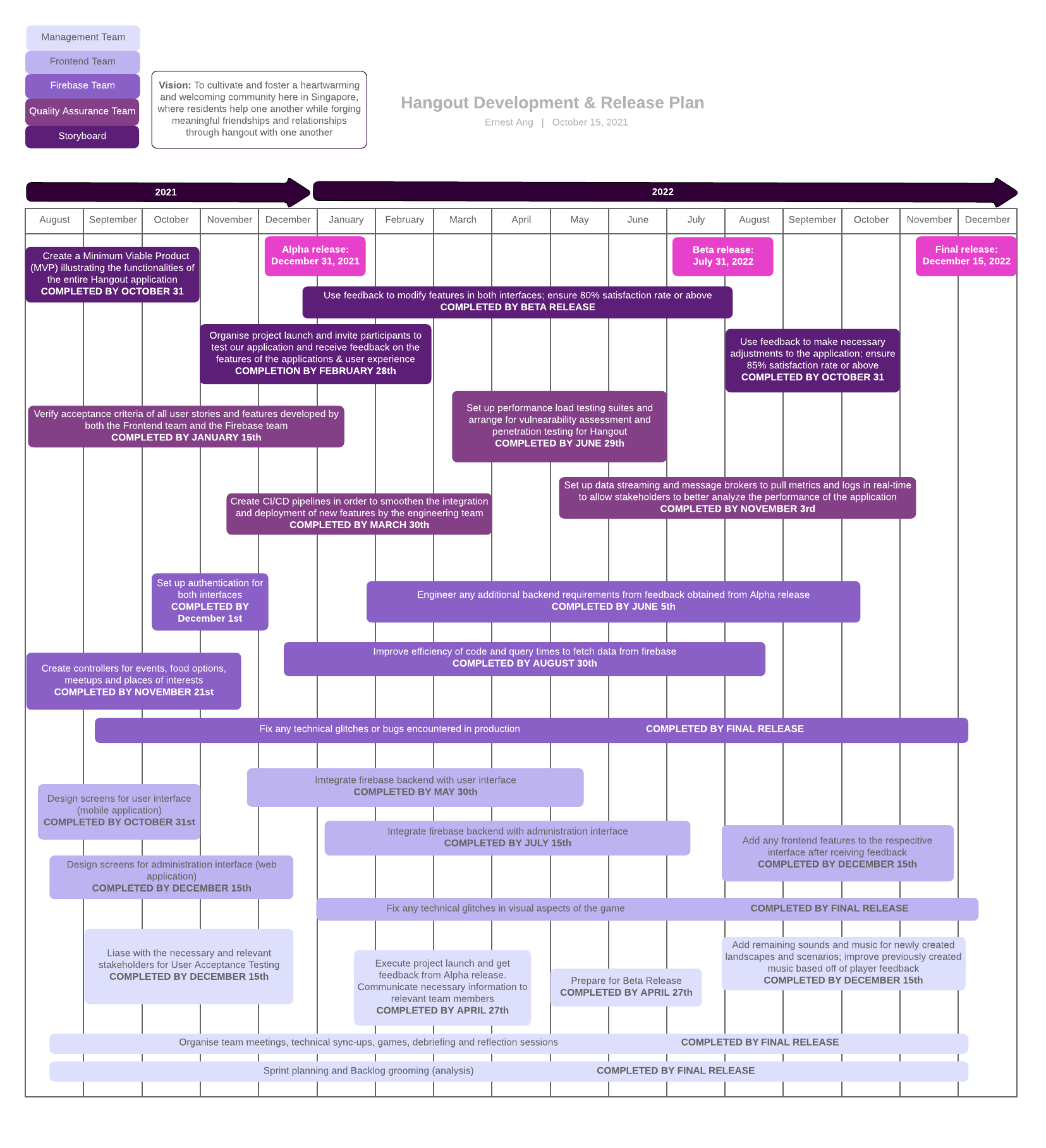
**5.2. Release Strategy**

We will be segmenting the overall delivery of HangOut 1.1.1 into three main releases. For each release, we will further break it down into multiple tasks and organise the timeline of completion of these tasks into weekly sprints in an Agile fashion. The project managers will estimate the difficulty and amount of time and resources required for each specific task and plan the appropriate and correct sequence of tasks to be completed in every sprint. This release strategy will allow room for our analysts managing the end-users as well as the project managers to gain sufficient feedback on the performance of the design product in incremental stages such that it would be easier to get these important pieces of information earlier to the designers and developers for feature correction, improvement and addition without slowing down sprints nor delaying the timelines in hitting milestones.

**5.2.1. Release Content**

We will have a total of three different releases for HangOut: Alpha release on 31st December 2021, Beta Release on 31st July 2022 and the Final release on 31st December 2022. We will soft launch the Alpha release of HangOutwhen some of the basic functionalities of HangOut have come together and are suitable to get initial feedback from the relevant stakeholders (e.g. Business Analysts, UI/UX designers). With this feedback, changes can be made before the complete product is put together and the Beta release of HangOut is launched, where most of the functional and non-functional requirements have been developed, tested, and verified by the Quality Assurance Team and is ready for the final User Acceptance Testing (UAT) arranged by the Project Management Team before the final launch of HangOut 1.1.1 into production.

**5.2.2. Release Schedule**



**5.2.3. Release Impacts**

Alpha Release launches early versions of HangOut where the majority of the functionalities are still in partial development. Feedback from this launch will most likely impact the project team, in particular, the engineering team consisting of the frontend developers engineering the mobile application and web application as well as the backend Firebase team, working to set up and connect the frontend to this service. Features may have to be modified slightly, changed completely or even removed based on the initial findings and reactions to this alpha release. The development and design team must stay focussed, open and Agile to changes to the subsequent project and design plan in accordance to the feedback received. Alpha release also sets the pacing as well as any delays or advancements to the timeline for completing and achieving key milestones. Hence, the main stakeholders and processes impacted by this release would be those in the project team itself.

Beta Release launches a near completed and tested version of HangOut . At this point, almost all of the features implemented have already gone through the entire software development lifecycle, from design to implementation to testing and integration. Stakeholders mostly involved with this release are the Quality Assurance and Software Testing Team in addition to the Frontend and Firebase Engineers. Working together using modern development processes such as Test Driven Development and Behavior Driven Development through user story cards, this version of the product is usually launched just 2 weeks prior to the actual final release. At this point, project managers will implement a code freeze to prevent the further possible introduction of any bugs to the system, with the exception of hot fixes and bug fixes. After this, final user acceptance testing is carried out by the project managers and business analysts in preparation of the final release. Similar to HangOut’s alpha launch, the main stakeholders and processes impacted by this release would be those in the project main team itself. In addition, clients as well as a selected group of end-users may be involved in user acceptance testing.

Final Release launches HangOut 1.1.1 into production, downloadable by individuals with smartphones running either on Android or iOS operating systems. Staff from the various Town Councils and Group Representation Constituencies will also be able to login to the Administration interface and begin to update food options, places of interests and happening events occurring in their respective regions and residents will be able to access these pieces of information from their mobile phones whilst having sophisticated and modern functionalities to sort through the thousands of restaurants, hidden gems, and exciting events occurring across the different regions of Singapore. This launch is the capstone event of the entire project, impacting mainly the end-users as well as the project managers and business analysts gathering feedback from the ground and continuing to engage their clients and users over the application. Bugs, major or minor, may surface though unlikely at this point and it is important for the engineering team to stay sharp and respond quickly by releasing hot fixes and minor patches in order to prevent frustration amongst the users. Ultimately, the main stakeholders impacted by this launch would be the project management team, the client, and the end users.

**5.2.4. Release Notification**

For the final release of HangOut 1.1.1, we will be using traditional and modern methods of advertising and notifying residents of the release in order to raise awareness of our application along with the product itself, through social media platforms and the various media outlets as well as recommendations on the respective application store (e.g. App Store for iOS and Play Store for Android). Subsequently, after users register for an account when using our application, any updates in releases will be done via email or text messaging with permission granted and enabled by the end users. We can also use custom in-app messaging to notify users on future new releases whenever there is a major, minor, or patch launch. For the administrative staff, we will be using their workplace email and their organisation as a medium to disseminate information regarding the launch of the administration interface in advance of the main user interface of HangOut to allow ample margin for the staff to create the current and upcoming events, food options and places of interests located in the vicinity of their region and group representation constituency.

For every release of HangOut , we will also be publishing the relevant documentation and the release notes for the current version. This will give a high level and concise overview on the purpose and functionality of the HangOut project while summarising the impact, issues highlighted, and bugs fixed from the previous version. Properly and well written release notes can also give the sales and marketing team content to share about with the clients and customers while also improving communication amongst stakeholders and team members, getting these important individuals aligned for each release and the one following it. Technical support crew can also leverage on release notes to guide end users who face difficulty navigating through the application or simply point the end users directly to the documentation itself for their perusal.