

Software Development Team
Wanderers
Nanyang Technological University
50 Nanyang Ave, Singapore 639798

Wanderers Release Plan

Version: 1.2
Last Modified: 03-25-2025

APPROVALS

Submitting Organization's Approving Authority:

Printed Name/ Position Table	Signature	Date	Phone Number
Kim Seo Jin / Project Manager	KSJ	03-25-2025	93723623

Wanderers Approving Authority:

Printed Name/ Position Table	Signature	Date	Phone Number
Isaac Chun / Release Manager	IZ	03-25-2025	88231156
Albert Yu / Development Lead	AY	03-25-2025	92372671

REVISION HISTORY

[illegible]

TABLE OF CONTENTS

1.	INTRODUCTION	6
2.	REFERENCED DOCUMENTS	6
3.	OVERVIEW	7
4.	ASSUMPTIONS, CONSTRAINTS, RISKS	7
4.1.	Assumptions.....	7
4.2.	Constraints	8
4.3.	Risks.....	9
5.	RELEASE APPROACH.....	10
5.1.	Rationale	10
5.2.	Release Strategy.....	10
5.2.1.	<i>Process</i>	<i>12</i>
5.2.2.	<i>Release Content.....</i>	<i>16</i>
5.2.3.	<i>Release Schedule.....</i>	<i>18</i>
5.2.4.	<i>Release Impacts.....</i>	<i>18</i>
5.2.5.	<i>Release Notification</i>	<i>20</i>
6.	GLOSSARY	21
7.	APPENDICES	22
7.1.1.	<i>Release Automation.....</i>	<i>22</i>

LIST OF FIGURES

Figure 1: Release Process Flow11

Figure 2: Release Process12

Figure 3: GitHub Release Artifact14

Figure 4: GitHub Changelog Artifact15

Figure 5: GitHub Release Workflow22

LIST OF TABLES

Table 1: Referenced Documents6

Table 2: Wanderers Risk9

Table 3: Release Content16

Table 4: Release Schedule18

Table 5: Release Impact18

Table 6: Release Notification Methods20

1. INTRODUCTION

The Wanderers Collaborative Travel Planning Platform, abbreviated as Wanderers, is a modern web-based application designed to streamline group trip organization and expense management efficiently. This Release Plan document outlines the comprehensive details of the platform, including its title, version number (1.0) and initial release. Its purpose is to guide users, developers and stakeholders through the release process, ensuring a seamless transition to utilizing Wanders for efficient travel planning. It also provides details about the strategy and considerations made by the team in releasing our system/updates in an organized and phased approach to ensure incremental updates, optimal user and system performance.

2. REFERENCED DOCUMENTS

Document Name	Document Number	Issuance Date
Project Plan	PP_v.1.2	03-12-2025
Risk Management Plan	RMC_v1.3	03-12-2025
Software Requirements Specification	SRS_v1.0.2	02-20-2025

Table 1: Referenced Documents

3. OVERVIEW

Wanderers' main purpose is to simplify the process of itinerary creation through features that enhance communication, provide transparent budget management and enable collaborative editing. The platform aims to address the challenges travelers face due to the fragmentation of different planning tools, ensuring a seamless and hassle-free travel experience.

The history of Wanderers' development includes the following key stages:

1. Research and Conceptualization

- Identifying the challenges and needs of travelers and then conceptualizing a solution that integrates various planning tools suitable for travelling into a single platform.

2. Design and Development

- Brainstormed and created the technical design and architecture of the platform from use case diagrams – incorporating features for communication, itinerary management and budget management.

3. Testing and Iteration

- Conducted internal testing to gather feedback, to provide incremental update to design and functionality and address any issues to user experience.

4. Deployment and Release

- Finalizing each iteration of the platform for release, ensuring all aspects/features as part of a specific version are well-built and testable, before being released.

4. ASSUMPTIONS, CONSTRAINTS, RISKS

4.1. Assumptions

As Wanderers is a platform that promotes collaborative editing and communication among individuals, **for a user** to fully experience the new features that Wanderers releases in each iteration, the following must be taken into consideration:

1. Users should have access to reliable internet connectivity and modern web browsers which support the platform's advanced features
2. Users should be familiar with basic digital navigation and how to interact with standard UI/UX flows including to the following example but not limited to (e.g Login, then confirming email through link)

Additionally, **external circumstances** that could impact the release of the system would be events such as:

1. Travel restrictions due to global events (Covid, or pandemic) – that could impact the usage and adoption of the platform

Wanderers

2. External API/deployment sources becoming unavailable, including the following:
 - Vercel – if this fails, the website will fail to deploy and update to the latest version, as well as make the current deployed website be inaccessible to users.
 - Google Places API – if this fails, users would not be able to seamlessly create itineraries, which heavily affects the release as most of our internal testing would fail.

Finally, **internal circumstances** that could impact the release would be events such as:

1. Lack of availability of skilled development and support staff
2. Lack of collaboration of multiple teams in Wanderers to ensure smooth deployment and implementation of Wanderers.

4.2. Constraints

Wanderers is an aspiring project, and it is indefinitely **bound by limited initial budget**, which limits the scope of initial features and marketing efforts, which may affect the reach and adoption rate of the platform. An example would be that Wanderers focuses most of its features in core aspects such as itinerary management and communication, but reduces its focus on custom profile management, which may cause lack of individuality in an app.

Schedule constraints in Wanderers could impact the timing of release, especially if unforeseen issues arise during our final testing phases, causing bugs or an incomplete release, or even a delayed release that might affect our relationships with stakeholders and our customers.

Internal and divisional involvement is also critical and crucial; any delays or lack of coordination between the Wanderers team could hinder the deployment process, and **external** system constraints such as compatibility issues with third-party applications or limitations of existing infrastructure may also impact Wanderers' releases due to failed deployments.

4.3. Risks

There are several risks associated with the release of Wanderers. The following table documents down each risk:

Risk	Description
Insufficient Testing	A major risk that impacts Wanderers release is due to insufficient testing in our platform. Insufficient testing may affect the usability of our services and major bugs to appear, negatively impacting user experience. Without testing, our platform would not be stable and perform as expected to our use cases and functional requirements.
Integration Issues with External APIs (Google Places)	Due to integration with external APIs, any changes or limitations to the API could impact the release due to the need to find workaround to our location-based services. Downtime to this API could also result in inability to test features and run our integration tests.
Difficulty in adopting the new platform	There is a risk of low user adoption if marketing efforts do not effectively reach the target audience. There could also be technical issues such as bugs or performance problems which could negatively affect user experience and satisfaction.
Dependency on Vercel for deployment	This dependency introduces risk related to service outages or performance issues on the Vercel platform. Since we rely on a host, there is no guarantee that international users would have the same experience due to input latency.

Table 2: Wanderers Risk

Unfortunately, mitigation strategies for these risks include robust testing and good quality assurance processes to identify and resolve these issues before release. Wanderers has contingency plans and identified risks will be documented in the projects Risk List and management in accordance with the Risk Management Plan.

5. RELEASE APPROACH

5.1. Rationale

The release approach for the Wanderers platform was established based on classic industry standards, and on thorough research, strategic planning and consideration of the various influencing factors highlighted by our assumptions, constraints and risks in Section 4. Lessons learnt from previous deployments of similar platforms also influenced the release approach for Wanderers such as phased rollouts to manage these risks effectively, to help shape a release strategy that prioritizes user experience, minimize disruption and ensure incremental and continuous improvement based on real-world usage and feedback.

5.2. Release Strategy

Wanderers will adopt a bi-weekly release cycle, utilizing feature branches, staging (stg) and production (prd) environments. This approach ensures that new features and improvements are regularly delivered to users while maintaining the high-quality standards highlighted in our Quality Management Plan. The branch structure for each release is depicted in the following image:

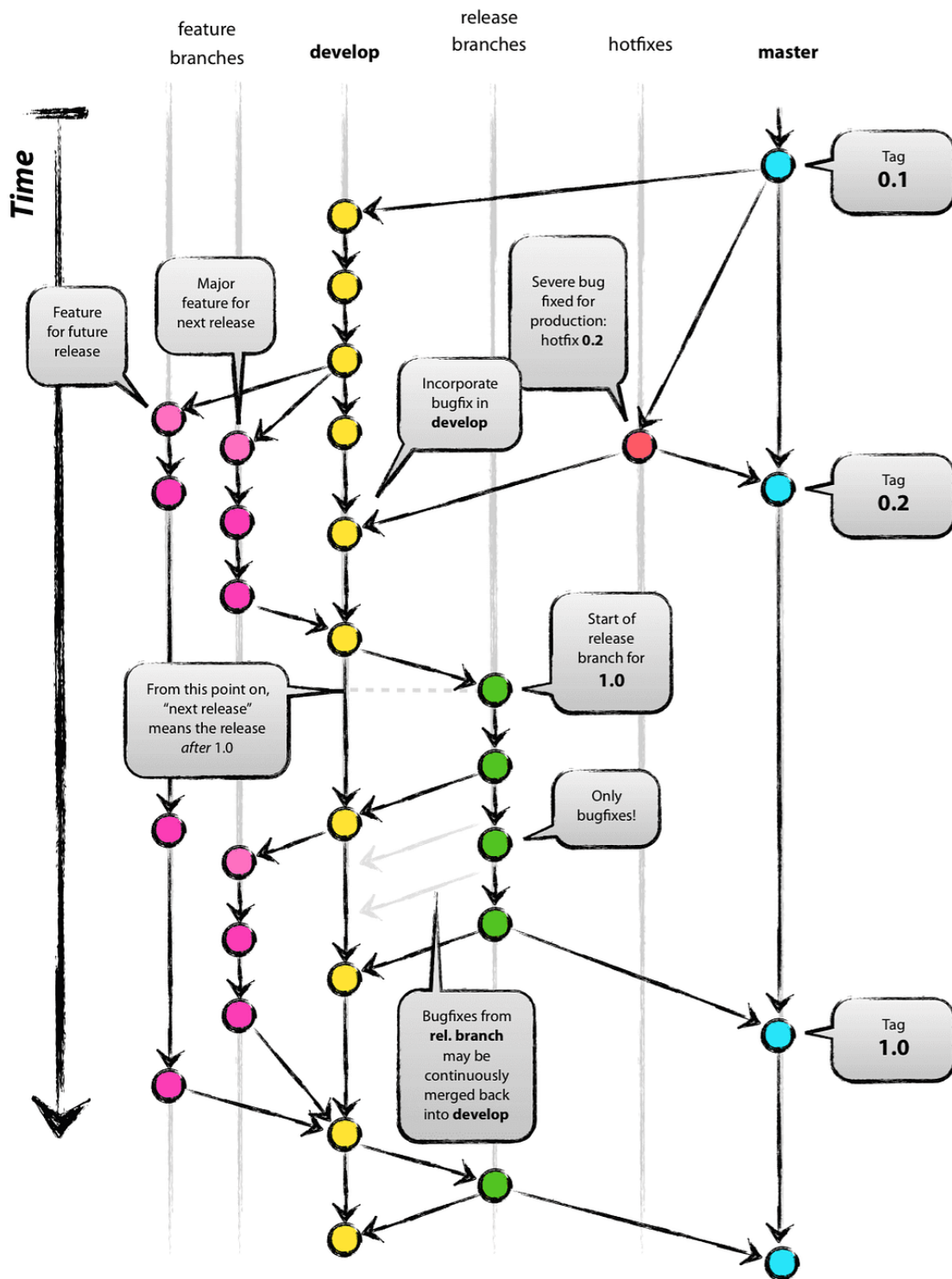


Figure 1: Release Process Flow

Wanderers

In the figure, the usage of **feature branch** allows for isolated development and testing of new functionalities, which reduces the risk of introducing bugs or performance issues into the main codebase.

Subsequently, **staging environment** is used by our QA team for thorough testing and quality assurance, ensuring that all new features and updates are fully vetted before being deployed to production. This environment serves as the critical checkpoint for identifying and resolving any issues that may arise during development. When the code base is ready for release, a release branch is created from this staging environment and the QA team uses this branch for testing and ensuring Wanderers is performing as expected. Any bugs or issues with code is highlighted to the **development** team and only bug fixes are made on the release branch until the product has been fully tested.

Finally, the **production environment** is where final and tested features are released to end users. Through maintaining separate staging and production environments, Wanderers can ensure smooth and reliable deployment process to minimize disruption and maximizing user satisfaction.

5.2.1. Process

More information about this process can be found the Appendix.

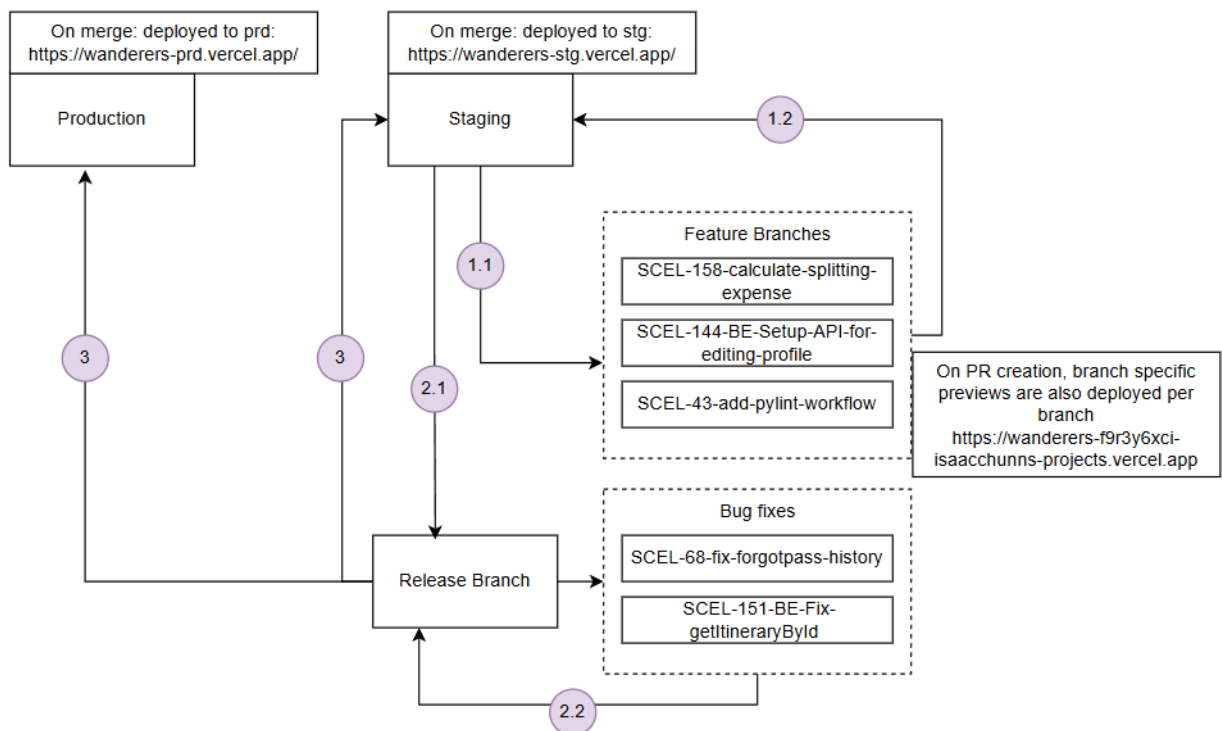


Figure 2: Release Process

The typical lifecycle of our release process goes in the following steps. The purple circle in the figure indicates the step number.

Pre-release:

1. Development of Features

- 1.1 The development team creates feature branches from the staging environment and conduct their developments according to our Release Content and Schedule, prioritizing the features that are essential for the next release.
- 1.2 After a pull request has been made on the feature and is approved by the development team on the approach, it is merged back into the staging environment. This process repeats until all features in the backlog have been exhausted, or a decision by the stakeholders that there are sufficient features in the release. At the same time, previews of the website are automatically deployed onto Vercel to show the current state of the product in that feature branch, as well as in the staging environment/website.

During release:

2. Release Branch Fixes

- 2.1 When the team is ready for a release, a release branch is created from the staging environment for our QA team to conduct their tests for that release. Developments on feature branch can still be done as the release branch is the state of the product being released.
- 2.2 In this cycle, only **bug fixes** can be done on the product in this branch and no longer any features development.

Finishing release:

3. Release to Production and Tagging

This process is fully automated using GitHub actions which does the following:


1. Update changelog
2. Commit changelog changes
3. Tag this particular commit
4. Push the commit
5. Make release on GitHub
6. Create PR to merge the release branch into the main branch


In this step, the progress and fixes from the release branch are merged into **stg and prd environments** after the QA team has stated that the product is in a stable state and agreed that the product can be released by the stakeholders. After the merge to each of the environments, the prd environment is tagged with the release version for clear benchmarks, as well as **automatic** updating of the changelog and release creation on GitHub. The stg and prd websites at <https://wanderers-stg.vercel.app/> and <https://wanderers-prd.vercel.app/> are automatically updated and deployed based on the codebase. The website that the customers directly interact with is <https://wanderers-prd.vercel.app/>.


GitHub Release:

- Automatic workflow causes automatic release artifact generation with changes made in this release on each merge to prd.

2 weeks ago

 isaacchunn

 v1.0.0

 39e57c3

Compare

v1.0.0 Latest

[1.0.0] - 2025-03-12

Added

- (BE) Other authentication related APIs (request-confirmation, confirm-account, forgot-password, reset-password)
- (BE) Ethereal implementation for sending emails
- (FE) Global user state (UserProvider and UserStore) with zustand implemented
- (BE) Authorization protection for non-authentication related backend routes
- (BE) Profile API (upload profile photo, update profile description and password change)
- (FE) Activity Creation Modal for Itineraries
- (FE) User Profile view to make changes to their profile information
- (FE) Middleware logic to redirect users appropriately based on user's logged in status
- (FE) Chat box socket to listen and emit messages to BE in itinerary pages (Only owners and collaborators can view)
- (BE) Chat box socket server to listen and emit messages to FE
- (BE/FE) Integration for authentication-related functions (register, login, confirm account, forget password)
- (BE/FE) Integration for Itinerary (CRUD itineraries)
- (BE/FE) Integration for Activity (CRUD activities)
- (BE/FE) Integration for User Profile
- (BE/FE) Integration for Chat feature in Itinerary


Fixed


- (BE) An issue with GET Itinerary only returning if they are the owner or collaborator regardless of visibility
- (BE) An issue with GET Activity returning deleted (non-active) activities
- (FE) An issue with the Create-Itinerary feature requires users to add at least one collaborator, restricting solo trip planning and reducing flexibility
- (FE) An issue where users are not redirected to "/" route from the Settings page

Changed

- (BE) Google Places API response to include more information for FE to use
- (BE/FE) Hardcoded server urls to use environment values instead
- (BE) Password change logic that prevents users from reusing their previous three passwords and ensures password adheres to best password practices

▼ Assets 2

 Source code (zip) 2 weeks ago

 Source code (tar.gz) 2 weeks ago




Figure 3: GitHub Release Artifact

Changelog Update:

- This changelog file in the repository is also automatically updated on each merge to prod using GitHub actions workflows.

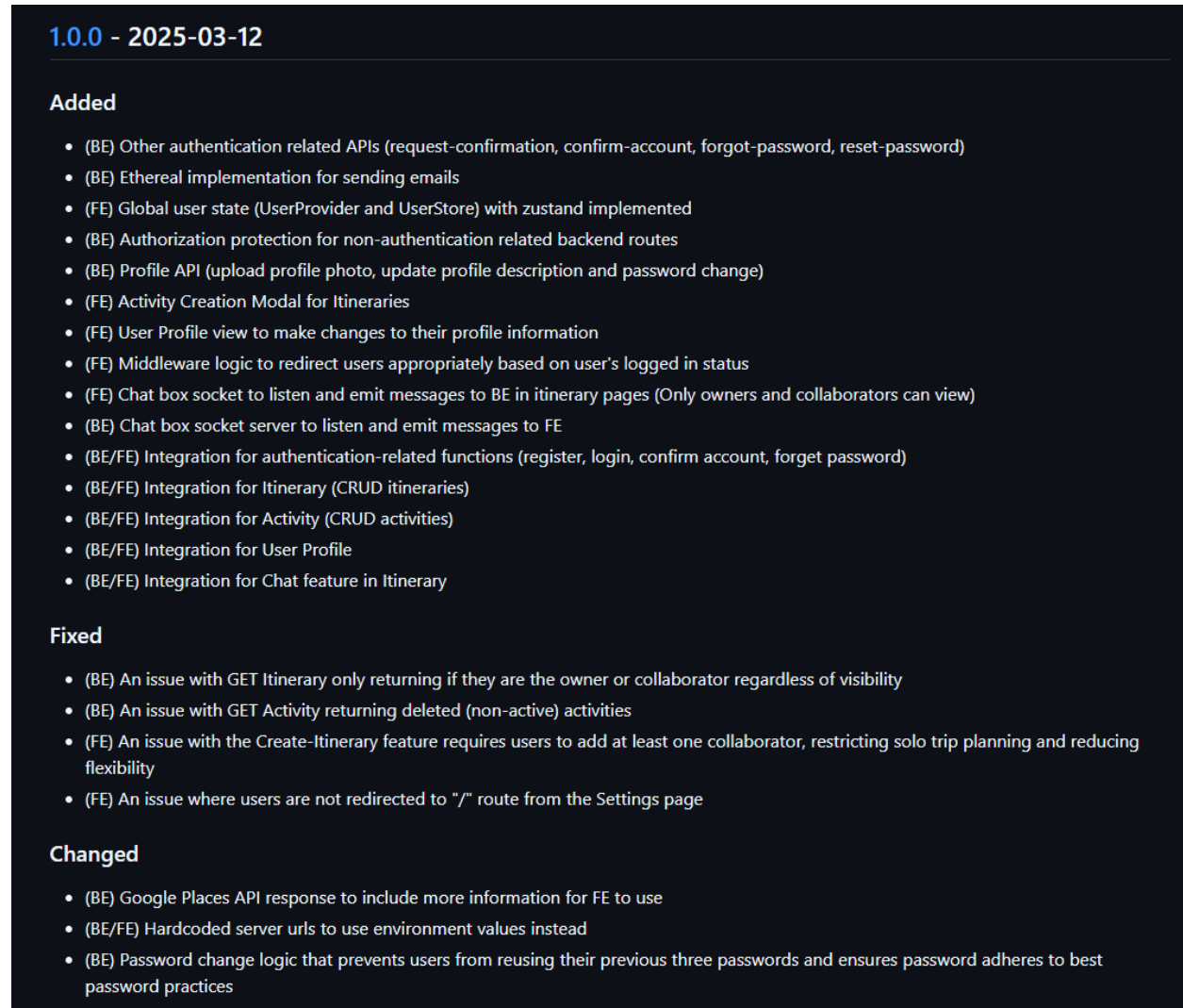


Figure 3: GitHub Changelog Artifact

In conclusion, our release process is iterative and incremental, following the workflow in Figure 1. Each time a new feature or release is planned, the cycle repeats with respect to Figure 1, allowing safe and easy incremental development of new features to Wanderers.

5.2.2. Release Content

Our release content is with respect to our tickets and planning done during our sprint ceremonies, as well as the lab guidelines. In each of the releases (two weeks), sufficient planning is done to ensure it is releasable while considering the risks that we might face as a team. The table below shows the planned content per release as part of our changelog that was maintained in the repository prior to the creation of the release plan.

Release	Content
v0.0.1	<ol style="list-style-type: none"> 1. Setup repository and coding guidelines <ul style="list-style-type: none"> - CODEOWNERS file to track mandatory reviewers of PRs - Contribution guidelines (feature branches, stg and prd environments) 2. Add changelog management 3. Add automatic script for updating the changelog
v0.0.2	<ol style="list-style-type: none"> 1. Full automation of release process
v0.0.3	<ol style="list-style-type: none"> 1. Add initial express API backend with signup and Login 2. Initial Nextjs project with TailwindCSS for frontend 3. Landing page for Wanderers Website 4. Add pylint workflow for checking lint compliance in PRs 5. Add release workflow for tagging prd branch and release creation
v0.0.4	Add: <ol style="list-style-type: none"> 1. (BE) CRUD routes for backend Itinerary API 2. (BE) CRUD routes for backend Activity API 3. (BE) Query Google Places API through backend (rate-limit protected) 4. (FE) Base layout web pages for Home, Login, Register, Itinerary viewing/creation and Activity viewing/creation 5. (FE) Itinerary handler for frontend to access backend Itinerary API for CRUD 6. (FE) Activity handler for frontend to access backend Activity API for CRUD and Google Places API 7. (FE) Mock Google Autocomplete for searching of activities in Itinerary page 8. (FE) Integration for viewing of Itineraries in Home page 9. CI/CD check for running tests and ESLint 10. Deployment to Vercel via Github Actions
v1.0.0 (first product release)	Add: <ol style="list-style-type: none"> 1. (BE) Other authentication related APIs (request-confirmation, confirm-account, forgot-password, reset-password) 2. (BE) Ethereal implementation for sending emails 3. (FE) Global user state (UserProvider and UserStore) with zustand implemented 4. (BE) Authorization protection for non-authentication related backend routes

	<ol style="list-style-type: none"> 5. (BE) Profile API (upload profile photo, update profile description and password change) 6. (FE) Activity Creation Modal for Itineraries 7. (FE) User Profile view to make changes to their profile information 8. (FE) Middleware logic to redirect users appropriately based on user's logged in status 9. (FE) Chat box socket to listen and emit messages to BE in itinerary pages (Only owners and collaborators can view) 10. (BE) Chat box socket server to listen and emit messages to FE 11. (BE/FE) Integration for authentication-related functions (register, login, confirm account, forget password) 12. (BE/FE) Integration for Itinerary (CRUD itineraries) 13. (BE/FE) Integration for Activity (CRUD activities) 14. (BE/FE) Integration for User Profile 15. (BE/FE) Integration for Chat feature in Itinerary
v1.1.0	<ol style="list-style-type: none"> 1. Error fixes of previous release 2. Add features based on user feedback

Table 3: Release Content

5.2.3. Release Schedule

The following table describes the schedule of our releases as well as the projected date that the release would be released to stakeholders.

Release	Projected Date (DD/MM/YY)
v0.0.1	27/01/2025
v0.0.2	29/01/2025
v0.0.3	01/02/2025
v0.0.4	19/02/2025
v1.0.0	12/03/2025
v1.1.0	25/03/2025

Table 4: Release Schedule

5.2.4. Release Impacts

Release	Impact
v0.0.1	<p>This initial setup of the repository and coding guidelines would establish the essential architecture of the system, impacting the development process through ensuring consistent coding standards and a well-organized repository structure by assigning reviewers to specific components. As such, code quality of the repository is enhanced and improves collaboration and onboarding efficiency.</p> <p>The addition of changelog management and automation streamlines future tracking of changes and updates which fosters transparency among the development team, making this release a strong foundation for a reliable and stable web application like Wanderers.</p>
v0.0.2	<p>The automation of the release process would significantly impact the business by reducing manual intervention which decreases the risk of human error in our release process. This will speed up deployment time, and ultimately lead to more frequent and reliable releases, which enhance the overall productivity of the Wanderers team. Consequently, end users will experience faster access and previews to new features and bug fixes, which improves the overall user experience.</p>
v0.0.3	<p>This release introduces the initial framework for our backend and frontend services, such as initial express API backend with signup and login functionalities, laying the groundwork for user management and authentication, essential for any web application. Furthermore, establishment of Nextjs for frontend allows Wanderers to be build on a responsive and modern design.</p> <p>To further increase our code quality, adding of pylint workflows would help Wanderers in the long run. The addition of the automated release workflow on</p>

	<p>GitHub would allow us to automate versioning and deployment and promote a structured release process, which is crucial.</p>
v0.0.4	<p>This release focuses on adding CRUD functionalities for backend itinerary and activity APIs, along with Google Places API integration in our backend, which significantly enhances the app's core features and provide users with a first look at comprehensive trip planning capabilities.</p> <p>Integration is also expected in this release to ensure seamless data flow and interaction between front and backend, and communication from the landing page for crucial actions such as login, registration, itinerary creation and activity viewing will be tested here as well.</p> <p>Finally, CI/CD checks and automatic deployment automation through Vercel would ensure that these features can be fully tested by the QA team, allowing previews of each development branch and lead to easier and more robust testing by our QA team and developers.</p>
v1.0.0	<p>This is the first product release which introduces the core features of Wanderers, including authentication-related APIs, request-confirmation, account management, itinerary handling and user management. This release would be the release with the most impact as future features and priorities would be prioritized based on this release due to the feedback from real-world users and enhancements. This release might also lead to changes in development strategy, release schedules and changes to existing protocols to accommodate changes to new system features and capabilities, which might make changes to system architecture.</p>
v.1.1.0	<p>This release intends to fix the issues identified from the previous release as well as add in feature requests from the user to allow Wanderers to dynamically change due to user requirements if the change is accepted by our control board. This release also signifies the start where the development team now deals with user requirements rather than internal requirements, which should increase the workload and environment of development.</p>

Table 5: Release Impacts

In conclusion, each release up to the main release is to incrementally update and deliver new features to refine Wanderers. From here on, each iteration would create emphasis on clearing any technical debt left by previous releases, addressing user concerns and needs, enhancing system architecture and robustness. Feedback from stakeholders would also be requested through timely communication and supported by the team based on updated documentation and available resources.

5.2.5. Release Notification

For each release, it is crucial to ensure effective communication with the relevant stakeholders to provide timely updates and necessary information. Stakeholders that require notifications includes the development team, our project manager, quality assurance team, end user and our investors.

For internal members of the Wanderers team, the latest developments and updates would be through the project's GitHub repository and Wiki pages that would be maintained internally by the team. On GitHub, information such as commit histories, hashes, tags, changelog updates, release artifacts would give the most up to date progress on Wanderers, while the wiki would also offer the up-to-date documentation and release notes of the app. Later, the **release notes** in the wiki would be published to a live website for Wanderers, so that end users are able to access the new changes to our application.

Notification methods vary across different users, based on the following table.

Stakeholder	Notification Method
Developers	GitHub, Wiki
Team Wanderers (Project Managers, QA Team)	GitHub, Wiki, Email
End Users	Email, Release Notes

Table 6: Release Notification Methods

6. GLOSSARY

API

- Application programming interface that allows creating, reading, updating and deleting objects within an application

Authentication

- The process of verifying the identity of a user or system, allowing access to protected resources based on credentials such as passwords or tokens

CI/CD

- Continuous Integration/Continuous Deployment; a set of practices and tools that enable automated testing, integration, and deployment of code changes to ensure high quality and frequent releases.

Middleware

- Software that acts as an intermediary between different systems or components, facilitation communication and exchange

CRUD

- Create, Read, Update, Delete: the four basic operations performed on database in a database or API.

7. APPENDICES

7.1.1. Release Automation

- The release is handled automatically via GitHub Actions. Upon confirmation by the QA team, the development/QA team would run the ‘Release new version of Wanderers’ workflow that automatically handles release generation and PR creation using Python.
- It has the following steps:

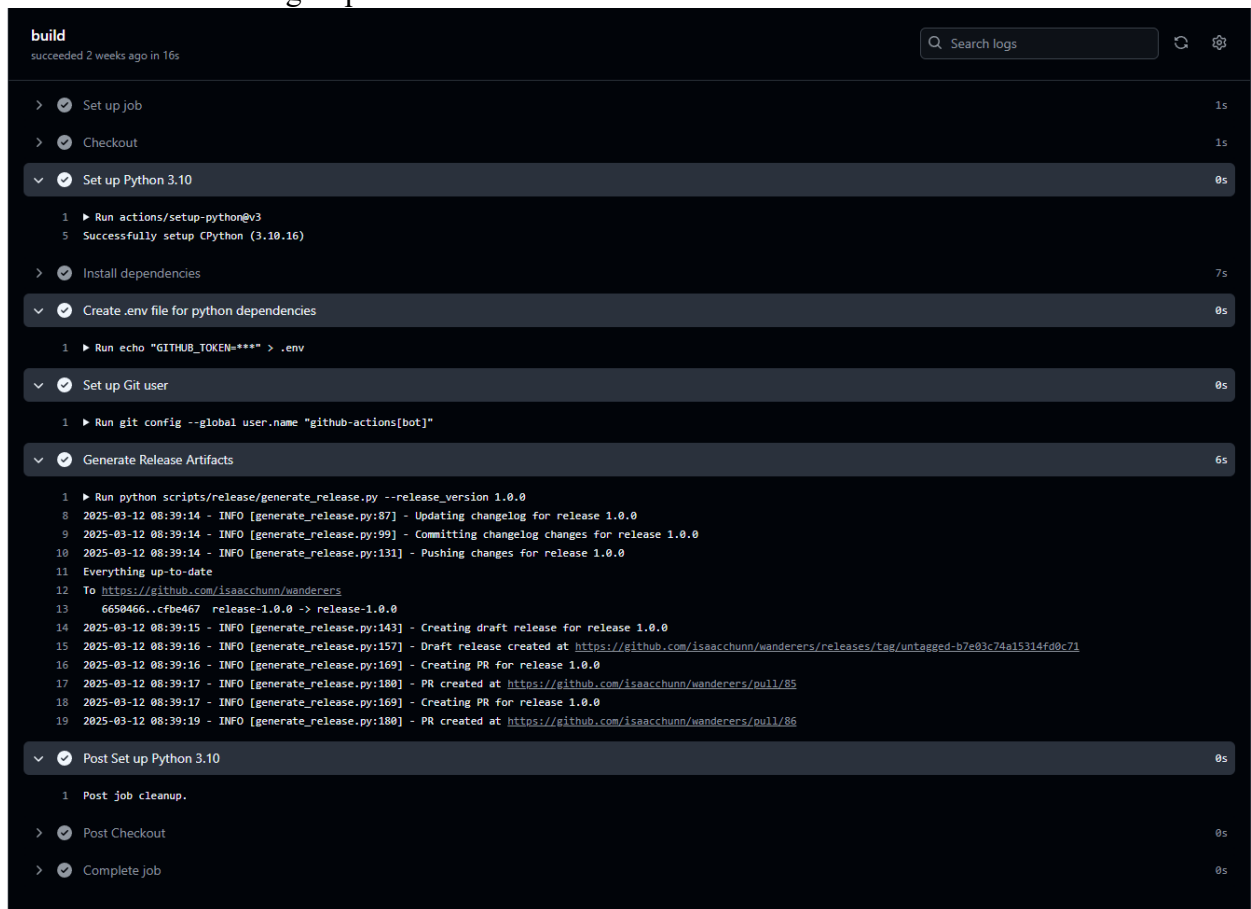


Figure 4: GitHub Release Workflow

- As a result, our release process is automated and more importantly, repeatable with minimal effort required – which boosts developer efficiency.