



|                   |   |
|-------------------|---|
| <b>Team</b>       | Bela Premchund, Sohum Singh, Mohammed Tehzeeb Govind, Kimera Pillay, Imaad Kajee, Altaf Ally, Kayla Shunmugam & Kayden Padayachee   |
| <b>SCRUM Link</b> | (Invite has been sent) – We do not have the ability to share the board  |
| <b>GitHub</b>     | <a href="https://github.com/ST10257468/Innovate-IT.git">https://github.com/ST10257468/Innovate-IT.git</a>   |
| <b>Quality</b>    | <a href="https://sonarcloud.io/project/configuration/GitHubActions?id=ST10257468_Innovate-IT">https://sonarcloud.io/project/configuration/GitHubActions?id=ST10257468_Innovate-IT</a> |
| <b>Security</b>   | <a href="https://app.snyk.io/org/st10257468/project/63a17567-89aa-465a-9908-9aa9a49132b5">https://app.snyk.io/org/st10257468/project/63a17567-89aa-465a-9908-9aa9a49132b5</a>         |
| <b>Website</b>    | <a href="https://umbilo-temple.onrender.com/">https://umbilo-temple.onrender.com/</a>   |
| <b>Minutes</b>    | <a href="https://app.clickup.com/9012876029/v/dc/8ckb4qx-1292">https://app.clickup.com/9012876029/v/dc/8ckb4qx-1292</a>   |

## Table of Contents

|                                    |    |
|------------------------------------|----|
| Introduction:.....                 | 3  |
| Work Agreement:.....               | 3  |
| Definition of Ready & Done:.....   | 9  |
| Roadmap: .....                     | 9  |
| Requirements:.....                 | 9  |
| User roles and Stories:.....       | 9  |
| Functional Requirements:.....      | 12 |
| Website:.....                      | 12 |
| App: .....                         | 13 |
| UX Journey Map .....               | 14 |
| Non-Functional Requirements: ..... | 17 |
| Analysis artifacts: .....          | 20 |
| Implementation documents:.....     | 33 |
| Testing:.....                      | 37 |
| Architecture artifacts:.....       | 37 |
| Design Patterns.....               | 37 |
| Cloud Architecture .....           | 42 |
| Security: .....                    | 43 |
| DevOps:.....                       | 44 |
| Running Costs:.....                | 45 |
| Change Management: .....           | 45 |
| Additional resources: .....        | 46 |
| References .....                   | 58 |
| AI Declaration:.....               | 61 |

## **Introduction:**

For this project, we chose the Umbilo Temple as our NPO since one of our members is does already donate and is familiar with the temple and their initiatives. We are aware of some of the issues regarding tourism and we hope that our website and app will publicize the temple. We created the website and app with the mindset of fixing these issues.

## **Work Agreement:**

### **Our Project:**

We are working with The Umbilo Temple, building both an app and a website for them. The Umbilo Temple is 150 years old, which is an unknown fact to many people. The aim of our project is to publicize the NPO, which may increase donations.

## **Members:**

### **Roles and responsibilities:**

#### **Bela J Premchund – Project Manager**

- The Project Manager will be the communication bridge between the client and the team. If there is anything that needs to be communicated to either the client or the team, the Project Manager will be the one to communicate this information.
- The Project Manager will be the one to assign tasks to each team member.
- The Project Manager will also assist the development and documentation team with their tasks and reviewing their work before it is marked as complete.

#### **Sohum Singh – Head Programmer**

- The Head Programmer will assist the development team with their tasks and will review and debug the code.
- If there are any situations where the Project Manager is unavailable, the Head Programmer will take charge.
- The Head Programmer will be testing both the website and the app before it is shown to the client.

### **Mohammed Tehzeeb Govind – Backend Developer**

- The Backend Developer will take lead of the development of the app and the database.
- The Backend Developer will create and connect the app to the database.
- The Backend Developer will assist with testing for the app.

### **Kayden Padayachee – Frontend Developer**

- The Frontend Developer will take lead in designing the mock-ups and the frontend design for the website.
- The Frontend Developer will also assist with the tests for the website.

### **Altaf Ally – Full-Stack Developer**

- The Full-Stack Developer will design the mock-ups and frontend design for the app.
- The Full-Stack Developer will also develop the backend code for the website.
- The Full-Stack Developer will assist with testing for the app.

### **Kimera Pillay – System Analyst**

- The System Analyst will create the functional and non-functional document.
- The System Analyst will also work with both the development and documentation teams for the domain analysis.
- The System Analyst will help to test the website.

### **Kayla Shunmugam & Imaad Kayee – Documentation**

- The documentation team will handle the documents that are created during the planning phase, i.e. the ERDs, the user roles and user stories.
- The documentation team will work with the development team to create the domain model diagram.
- They will also record all changes that are made throughout the project.
- The documentation will conduct the first line of tests on the app.

## **Communication:**

### **Email:**

- For emails to the client, all emails will be sent by Bela with Sohum being copied in for all emails, while the rest of the team will be blind copied in for specific emails. For example, if it is about an event, the whole team will be copied in whereas if it is for an update from the client, Sohum will be copied in.
- Alternative contact information (i.e. mobile number) will also be provided at the end of the email.

### **Meetings:**

- Virtual meetings will be held either on MS Teams or on WhatsApp.
- Meetings will be arranged at least a day before the meeting date. If an emergency meeting is required, the team will be notified at least 2 hours prior to the meeting.
- All in-person meetings are arranged a week before the meeting and on a business day.
- Emergency meetings include meetings to discuss major changes or faults that need immediate attention or meetings that are set by the client due to their schedule.

## **Working Hours:**

- All communications and meetings will be conducted between the hours of 8am and 7pm during business days. No meetings will be arranged on a weekend or public holiday unless it is an emergency meeting.
- For weekends and public holidays no work will be assigned, if there are any tasks that need immediate attention, the team would only need to work between the hours of 8am to 12pm.
- The same applies to birthdays. If a team member's birthday falls on a business day, no tasks will be due on that day. If there are tasks that need immediate attention, the team member would only need to work between the hours of 8am to 12pm.

## **Decision making:**

- Any decisions that need to be made are first discussed with the whole team. The team will then vote on which solution they think is the best and are comfortable with doing,

- All decisions about the project will be communicated to the client before we continue. If there are any doubts or they are not comfortable with our solution, it will be communicated back to the team, and we will all discuss a new solution.

## **Workload Distribution:**

### **Website Development:**

- Altaf Ally: 40%
- Kayden Padatachée: 40%
- Bela Premchund: 20%

### **App Development:**

- Mohammed Tehzeeb Govind: 45%
- Altaf Ally: 30%
- Sohum Singh: 15%
- Bela Premchund: 10%

### **Documentation:**

- Kimera Pillay: 25%
- Kayla Shunmugan: 30%
- Imaad Kajee: 30%
- Bela Premchund: 15%

### **Testing:**

- Altaf Ally: 20%
- Kayden Padayachee: 20%
- Sohum Singh: 45%
- Bela Premchund: 15%

## **Timelines:**

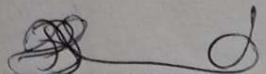


| Task Name            | Duration       | Start               | Finish              |
|----------------------|----------------|---------------------|---------------------|
| <b>Umbilo Temple</b> | <b>87 days</b> | <b>Thu 25/07/03</b> | <b>Fri 25/10/31</b> |
| Planning             | 14 days        | Thu 25/07/03        | Tue 25/07/22        |
| Documentation        | 45 days        | Wed 25/07/23        | Tue 25/09/23        |
| App Development      | 44 days        | Wed 25/07/23        | Mon 25/09/22        |
| Web Development      | 55 days        | Wed 25/07/23        | Tue 25/10/07        |
| Testing              | 16 days        | Mon 25/09/22        | Mon 25/10/13        |
| Presentation Prep    | 15 days        | Mon 25/10/13        | Fri 25/10/31        |

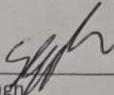
### **Conflict resolution:**

- Members are encouraged to voice their concerns.
- If there are any conflicts, it is raised to either the Project Manager or the Head Programmer.
- All conflicts are handled between the concerned parties and the Project Manager, or the Head Programmer will be the mediator.
- Conflicts will be addressed immediately and a solution, that all concerned parties agree to, will be implemented.

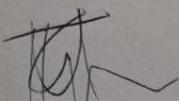
## Signatures:



Bela J Premchund  
ST10257468  
Project Manager



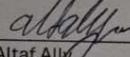
Sohum Singh  
ST10257909  
Head Programmer



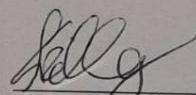
Mohammed Tehzeeb Govind  
ST10267459  
Backend Developer



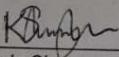
Kayden Padayachee  
ST10385543  
Frontend Developer



Altaf Ally  
ST10254661  
Full-Stack Developer



Kimera Pillay  
ST10250904  
System Analyst



Kayla Shunmugam  
ST10132758  
Documentation



Imaad Kajee  
ST10050486  
Documentation

## Definition of Ready & Done:

### Definition of Ready:

For each task, our definition of ready is where the task is ready to be reviewed by either the Project Manager or the Head Programmer. For example, the app would be defined as ready when it has all the features needed and can be tested.

### Definition of Done:

Our definition of done is where the task can be marked off as completed. For example, once the app is tested and changes have been made according to feedback, the app is considered to be done and will be published.

## Roadmap:

|   | i | Task Name         | Duration | Start        | Finish       | Predecessors |
|---|---|-------------------|----------|--------------|--------------|--------------|
| 1 |   | Umbilo Temple     | 87 days  | Thu 25/07/03 | Fri 25/10/31 |              |
| 2 |   | Planning          | 14 days  | Thu 25/07/03 | Tue 25/07/22 |              |
| 3 |   | Documentation     | 45 days  | Wed 25/07/23 | Tue 25/09/23 | 2            |
| 4 | 📅 | App Development   | 44 days  | Wed 25/07/23 | Mon 25/09/22 | 2            |
| 5 | 📅 | Web Development   | 55 days  | Wed 25/07/23 | Tue 25/10/07 | 2            |
| 6 | 📅 | Testing           | 16 days  | Mon 25/09/22 | Mon 25/10/13 |              |
| 7 | 📅 | Presentation Prep | 15 days  | Mon 25/10/13 | Fri 25/10/31 |              |

## Requirements:

### User roles and Stories:

Identified users: Admin users and normal users.

#### Administrator User Stories

User Story: As an Administrator, I want to log into a secure, password-protected area so that I can manage app content safely and prevent unauthorised access.

- Priority: High
- Estimation: 5 story points
- Sprint: 1

- Status: Implemented
- Non-Functional Requirements:
  - Security: Ensure strong authentication mechanisms (e.g., password policies, potential integration with multi-factor authentication).
  - Reliability: System must be available 99.9% of the time for administrators to manage content.

User Story: As an Administrator, I want to update the "gallery" section so that I can ensure users have accurate and up-to-date information.

- Priority: Medium
- Estimation: 3 story points
- Sprint: 1
- Status: Implemented
- Non-Functional Requirements:
  - Maintainability: Updates should not require extensive downtime; changes should be logged for audit purposes.

User Story: As an Administrator, I want to add, remove, or update gallery images and event details so that I can keep the community informed about recent activities and upcoming events.

- Priority: Medium
- Estimation: 4 story points
- Sprint: 2
- Status: Implemented

Non-Functional Requirements:

- Scalability: The system should efficiently handle increased image uploads as the gallery grows.
- Reliability: Ensure the gallery updates without data loss and is accessible consistently.

## User Stories

1. User Story: As a Donor, I want to view banking details and certification information so that I feel confident about my donation's legitimacy and impact.

— Priority: Medium

○Estimation: 2 story points

○Sprint: 1

○Status: Implemented

○Non-Functional Requirements:

■Maintainability: Banking information must be easily updatable without extensive system downtime.

3. User Story: As a User, I would like to place a booking for a prayer.

— Priority: Medium

○Estimation: 2 story points

○Sprint: 2

○Status: Not yet implemented

○Non-Functional Requirements:

■Reliability: Notifications must be sent promptly upon successful

4. User Story: As a Public User, I want to browse the "About Us" section so that I can understand the Arya Samaj mission and values.

● Priority: Medium

● Estimation: 2 story points

● Sprint: 1

● Status: Implemented

● Non-Functional Requirements:

- Scalability: The content should be easily scalable to accommodate additional information as needed.

User Story: As a Public User, I want to view recent events and photos in the gallery so that I can stay connected to the organization community and activities.

- Priority: Medium
- Estimation: 3 story points
- Sprint: 2
- Status: Implemented
- Non-Functional Requirements:

- Reliability: The gallery must load quickly and efficiently, even during high traffic periods.

5. User Story: As a Public User, I want to review the "Privacy" page so that I understand how my data will be protected and used.

- Priority: Medium
- Estimation: 1 story point
- Sprint: 1
- Status: Implemented
- Non-Functional Requirements:

Security: Ensure that the privacy policy complies with applicable data protection regulations and is reviewed regularly

## Functional Requirements:

### Website:

#### **About, History, and Community Impact**

On the About page , readers will find an interactive timeline component that depicts the temple's history over the years. By clicking on certain milestones, visitors can view accompanying photographs and brief informative descriptions, immersing themselves

in the Umbilo Temple's rich history. A dedicated "Community Initiatives" page will showcase ongoing outreach programs, such as school collaborations and food drives, with visible progress indicators (for example, "75% toward our goal of 1,000 meal packs"). To increase participation, site visitors can download a detailed PDF "Temple Profile" pack, which describes the temple's history, mission, and notable accomplishments in a single, professionally created document.

## **Home and Landing Experience**

The Umbilo Temple platform's Home Page will greet visitors with a dynamic banner highlighting forthcoming key events and announcements. This carousel will cycle through at least three high-resolution highlights with smooth fade transitions, keeping users up to date on the newest temple events. A sticky top menu will remain visible as visitors browse, offering easy access to important sections such as Home, About, Events, Donations, and Contact, ensuring intuitive and smooth navigation. In addition, a real-time "What's On Today" widget will display any ceremonies or gatherings scheduled for the current date, along with direct links to comprehensive event pages for easy research.

## **Gallery**

Admin users will be able to add images and videos to the gallery that will be visible to the public. There is a specific page where users can view the gallery containing all images and videos from different events.

## **App:**

### **Donations**

For this, users will find the NPOs banking details where they can then copy and place into their personal banking apps. Admin users will be able to add, delete and edit banking details that is shown on the app.

## **Event Management**

The activities page will list all upcoming and past temple activities, with filters for categories like "Religious Ceremony" and "Community Outreach" and a keyword search for quick discovery. Each event will have linked maps with location specifics, comments on dress code and required offerings, and a gallery of associated images and videos, resulting in a full, informative preview of every temple activity.

## **User Profiles and Personal Dashboard**

Registered users will have access to a unified Personal Dashboard that consolidates their major interactions, including prayer bookings. Individuals can use this dashboard to change their profile information (including name, and contact information). This customizable hub allows each user's temple activity to be conveniently tracked and controlled in one area.

## **Role-Based Content Management (Admin)**

Committee members can use the Admin Dashboard, which includes rich-text editors and media upload interfaces, to add or amend pages, events, gallery pieces, and announcements. The platform will allow structured approval workflows: when non-admin users submit new content, such as community tales, administrators must review and publish them. This ensures that all public-facing information remains correct and in line with the temple's standards. Admin users do have the ability to make a user an admin.

## **Gallery**

Users can upload images and video to the app, they will then be approved by the admin before the image is added to the public gallery.

## **Bookings**

Users can make a booking for an event or prayer, these bookings will be approved or denied by the admin user. If a booking is denied, users will be provided with a reason as to why. Users will also be able to view the status of their bookings and a record of all the bookings they have made.

## **UX Journey Map**

### **User Persona**

Name: Mr. Rajen Pillay

Age: 67

Occupation: Retired accountant

Technology Comfort Level: Moderate – uses a smartphone but sometimes feels unsure with new apps.

**Goals:** To stay updated on temple activities, make donations online, and remain connected with the temple community.

**Pain Points:** Finds small text and busy layouts difficult to read. Gets frustrated if the website is slow or instructions are unclear.

**Motivation:** Deep spiritual commitment and a wish to stay active in temple life while enjoying the convenience of online access.

| User Journey Map  |  |   |   |   |   |
|-------------------|--|---|---|---|---|
|                   | AWARENESS  | LOGIN / REGISTRATION                        | EXPLORE HOME DASHBOARD                          | PARTICIPATE / DONATE                          | CONTINUED ENGAGEMENT                    |
| <b>OBJECTIVES</b> | Discover the temple's new web app.                     | Create an account or log in.                | Navigate temple info and updates.               | Request temple visit or donate online.        | Receive reminders and updates.          |
| <b>NEEDS</b>      | Wants easy access to temple updates.                   | Simple sign-up with clear instructions.     | View upcoming events and contact details.       | Secure, easy payment processing form.         | Notifications about events and prayers. |
| <b>FEELINGS</b>   | Curious but unsure about using technology.             | Slightly anxious; struggles with passwords. | Interested but sometimes lost.                  | Proud and happy to contribute.                | Grateful and satisfied.                 |
| <b>BARRIERS</b>   | Doesn't know where to find the link or how to open it. | Small text and confusing button placement.  | Overwhelmed by too many sections on one screen. | Show loading page under confirmation message. | Doesn't always get updates on time.     |

User Journey Maps give an overview of the customer experience. How do you want your business to reach users?

| LET'S END THE SESSION WITH ACTION ITEMS |               |
|---|---------------|
| ACTION ITEM                             | POINT PERSON  |
| Simplify login and enlarge fonts        | UX Designer   |
| Add event reminder system               | Developer     |
| Include visual feedback after donations | Front End Dev |
| Create home screen quick-access icons   | UI Designer   |

## User Experience Journey Map

| Stage                     | User Goals                                    | User Actions  | User Emotions                                    | Challenges                                       | Opportunity for Improvement   |
|---------------------------|---|---|--|--|---|
| 1. Awareness & Access     | Open the Umbilo Temple web app                | Launches the website on phone or tablet                                   | Interested but a little unsure about the process | Sometimes struggles to remember the correct link | Provide a clear QR code at the temple or an easy home-screen shortcut     |
| 2. Login / Authentication | Log in quickly and without stress             | Types in email and password on the "Welcome Back" screen                  | Nervous at first, then relieved once logged in   | Can forget the password or press the wrong key   | Add a "Remember Me" feature and a large "Forgot Password" button          |
| 3. Home Dashboard         | Understand where everything is                | Reads the welcome message and sees buttons for Events, Donate, and Logout | Calm and confident                               | Buttons may be small or close together           | Use bigger, high-contrast buttons with clear icons                        |
| 4. Viewing Events         | Find out about upcoming temple events         | Clicks "Events" to open the calendar                                      | Happy and interested                             | Some text on the calendar may appear small       | Allow zoom-in or voice-assisted reading for accessibility                 |
| 5. Requesting a Visit     | Book a suitable date and time to attend       | Selects preferred day and time for temple visit                           | Satisfied and spiritually fulfilled              | Unsure if the request went through successfully  | Show a confirmation or "Thank You" pop-up after submitting                |
| 6. Making a Donation      | Donate easily and know the temple received it | Opens "Donate" page, reads EFT instructions, adds reference details       | Proud and peaceful                               | No confirmation that the donation was received   | Provide an automated thank-you message or email receipt                   |
| 7. Logout & Reflection    | Leave the app safely                          | Clicks "Logout"   | Relaxed and None content                         | None significant                                 | Display a short appreciation message such as "Thank you for your support" |
| 8. Continued Engagement   | Stay connected to temple life                 | Returns weekly to check events or donate again                            | Loyal and happy                                  | Sometimes forgets to check the app               | Add optional email or app notifications for upcoming events               |

## Overall Experience Summary

The Umbilo Temple Web App delivers a simple and meaningful online experience for older users. Its straightforward layout helps reduce confusion, and all key functions—events, donations, and logout—are easy to find.

To make the experience even better, small updates such as clearer confirmation messages, slightly larger fonts, and optional reminders could help users feel more confident and valued.

Overall, this journey shows how thoughtful design can make digital participation more accessible for elderly members of the community.

## Non-Functional Requirements:

### Performance and Scalability

The Umbilo Temple platform will be designed to enable consistently fast and responsive interactions, allowing users to access primary pages and important API endpoints with no discernible latency under typical operating conditions. To handle traffic surges, such as during holidays or big temple events, the design will include distributed caching layers, intelligent load balancers, and optimized query patterns. Caches will keep frequently accessed content on both the edge and in-memory levels, lowering backend load and improving retrieval times. Database schemas will be thoroughly indexed and read replicas will be employed to move analytics and reporting queries away from the core data store. On the mobile front, background synchronization mechanisms will adjust polling intervals based on real-time network bandwidth and device battery levels, finding a balance between timely data updates and resource conservation to provide a seamless, always-on experience. (GeeksforGeeks., 2025; TechTarget, 2025)

### Reliability and availability

To maintain the trust and expectations of the Umbilo Temple community, all important service components will be deployed across redundant servers and different availability zones. Automated health-check procedures will continuously monitor application endpoints, queue sizes, and database connectivity, prompting zero-downtime failovers if anomalies are discovered. This design ensures that the platform is accessible even in the event of hardware failures, network outages, or regional disruptions. Regular backup schedules and snapshot procedures will preserve transactional and media data while retaining it for recovery and auditing needs. Detailed disaster recovery plans, including runbooks for manual intervention, will be developed so that if an automated failover

fails to fix a problem, the operations team may quickly restore full functioning with minimal service disruption. (GeeksforGeeks., 2025; TechTarget, 2025)

## **Security and Privacy**

Security will be built into all layers of the system, from transport encryption to access administration. All connections between users, mobile clients, and servers will be safeguarded by industry-standard TLS protocols, while sensitive data at rest—such as donor information, volunteer records, and user credentials—will be encrypted using powerful algorithms. A strong role-based access control system ensures that only authorized staff can execute administrative tasks, while all authentication attempts and configuration modifications are recorded in an immutable audit trail for forensic analysis. Passwords will be hashed with a tried-and-true algorithm and salted to withstand brute-force attacks, and users can opt into multi-factor authentication routines for enhanced account security. Clear consent prompts, user-facing privacy settings, and adjustable data lifecycle policies that determine how long personal information is maintained before secure archiving or erasure will all be included as privacy measures in accordance with local requirements. (GeeksforGeeks., 2025; TechTarget, 2025)

## **Usability & Accessibility**

The platform's user experience will be characterized by a focus on intuitive design and broad accessibility. Form validations, dynamic menus, and content sliders are examples of interactive features that provide rapid visual or textual feedback to users, allowing them to understand system responses with minimal cognitive friction. All interfaces will use responsive design concepts, which will automatically modify layouts, font sizes, and touch-target widths across devices ranging from smartphones to desktop monitors. Adherence to recognized accessibility standards ensures that all templates and components include keyboard-only navigation, screen-reader support, and colour-contrast rules. Furthermore, specific feedback mechanisms—such as in-app surveys and context-sensitive help prompts—encourage users to share usability insights, allowing improvements to be prioritized based on actual user needs and experiences. (GeeksforGeeks., 2025; TechTarget, 2025)

## **Maintenance and Supportability**

Long-term maintainability will be built on a clear, modular code structure, with each functional area separated into individual components or microservices. Source code will adhere to generally established design concepts such as separation of concerns,

consistent naming standards, and dependency inversion, allowing developers to easily locate, expand, or refactor specific functionality without unintended consequences. Every major module will be accompanied by comprehensive documentation, such as API specifications, architectural diagrams, and inline code comments, as well as powerful automated test suites to evaluate functionality on each contribution. Continuous integration pipelines will perform unit, integration, and end-to-end tests to detect regressions early in the lifecycle. (GeeksforGeeks., 2025; TechTarget, 2025)

### **Compatibility & Portability**

To provide broad accessibility and reduce obstacles for end users, the web application will be certified against the most recent long-term-support releases. Mobile applications will target popular operating system versions and be designed to function smoothly on devices with various resource profiles; adaptive feature flags will prevent advanced animations or background services when loaded on low-end hardware. Containerization technologies will be used throughout the development and deployment process to encapsulate services and dependencies in portable runtime environments. This technique not only ensures consistency between development workstations, staging clusters, and production clouds, but also enables the Umbilo Temple team to select or migrate between hosting providers without requiring significant modifications. (GeeksforGeeks., 2025; TechTarget, 2025)

### **Compliance & Legal**

- Financial donations, personal information, and transactional logs shall all be managed strictly in compliance with applicable legal and regulatory frameworks. The system's auditing capabilities will generate detailed records—complete with timestamps, user identification, and operation details—to meet both internal governance and external auditing standards. Before registering or participating in sensitive features, users will be requested to review and expressly accept the most recent Terms of Service and Privacy Policy materials. A version-tracking method will keep track of the policy version that corresponds to each user's consent. To maintain security certification and compliance, the platform will undergo regular vulnerability assessments and third-party penetration testing, with any high-severity discoveries handled within an agreed-upon repair window. (GeeksforGeeks., 2025; TechTarget, 2025)

## **Localization and Internationalization**

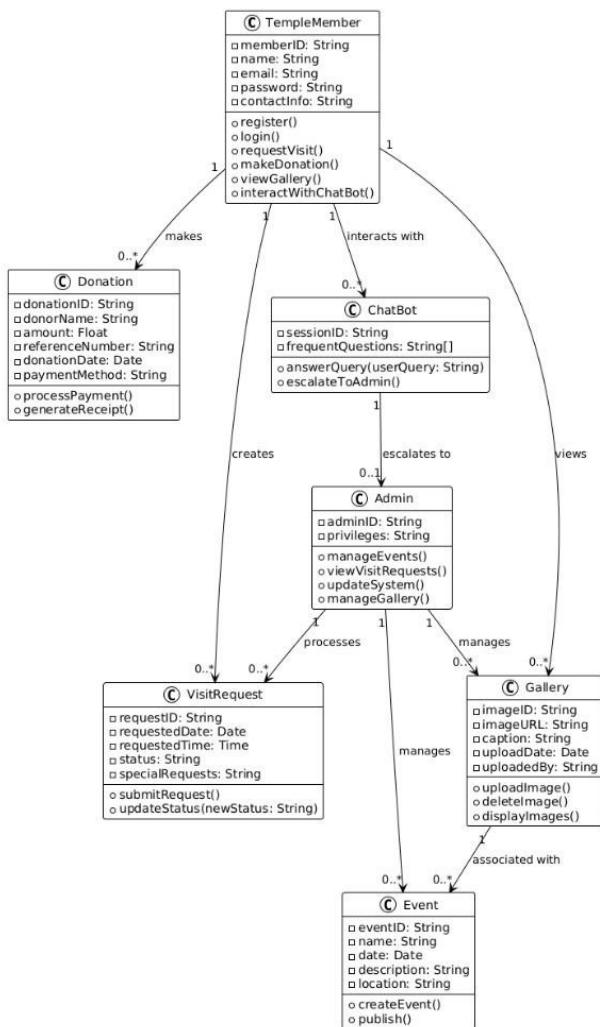
The Umbilo Temple platform will be designed from the ground up with localization in mind, externalizing all user-facing text, labels, and messages into resource bundles that can be translated without affecting the core software. Date formats, numeric conventions, and currency symbols will be automatically aligned with the user's locale settings, and user preferences can overrule system defaults as needed. This architecture will make it easier to introduce new languages—such as Afrikaans or Portuguese—as well as cultural adaptations in the future, such as support for right-to-left scripts or locale-specific visual themes, ensuring that every visitor feels at home, regardless of language or culture. (GeeksforGeeks., 2025; TechTarget, 2025)

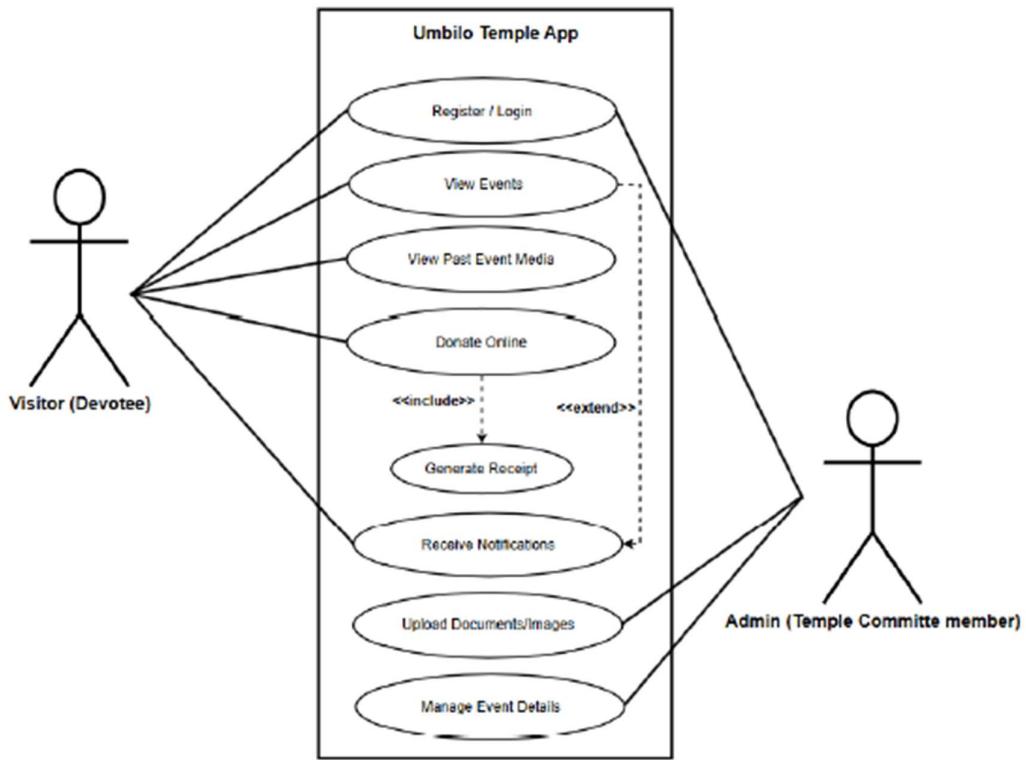
## **Analysis artifacts:**

The design phase of the Umbilo Temple mobile application is a key link between system analysis and implementation. It turns abstract concepts into real, functional, and user-focused design plans. This part shows the design artifacts that show how the app's architecture, parts, and user interactions were set up to fulfill the needs of the Umbilo Temple and the people that live there. (GeeksforGeeks, 2025) These design models give a clear picture of how the system is logically and physically built, making sure that everything, from the user interface to the backend integration, meets the project's functional and non-functional needs. The Umbilo Temple App was made to make it easier for people to get involved in the community and run the temple using current technologies. The app lets people see event schedules, make safe donations, learn about the history and culture of the area, and get reminders about forthcoming events. For administrators, it lets them manage event content, add media, and amend announcements through secure access. (GeeksforGeeks, 2025) The system's layered architecture was based on this dual-user approach, with usability, scalability, and security as the main goals for all modules. The development team used important software engineering ideas like Domain-Driven Design (DDD) to make sure that the technical structures matched how the temple works in the real world, and Model-View-Controller (MVC) architecture to keep the interface, business logic, and data handling separate. (AWS, 2025) Also, the design focuses on modularity so that future improvements can be made, including adding cloud-based storage and donation tracking systems. Use Case Diagrams, Sequence Diagrams, System Architecture Diagrams, and UI Wireframes are all examples of diagrams and models that help you see how the system works and how it flows. Use case diagrams show how users engage with the system and what the system's goals are. Sequence diagrams show how operations like donations and event uploads logically proceed. The architecture

diagram shows how the frontend app, backend API, and database layers all link to each other. (GeeksforGeeks, 2025) These artifacts work together to give a full picture of the system architecture, making sure that everyone involved, from developers to the NPO committee, has the same understanding of how the app works and is built.

The design artifacts are not only useful for implementation, but they also help make sure that the quality of the job is high. (GeeksforGeeks, 2025) The design makes sure that the Umbilo Temple App stays easy to use, efficient, and long-lasting by clearly showing how people, processes, and data are related to each other. These artifacts are a reflection of the project's main goal: to combine technology with tradition so that the temple can reach more people and run its business better in the digital age. (Interaction Design Foundation, 2024)





### **1. Visitor (Devotee):**

Represents the average person who uses the app to find out about forthcoming events, look at images and videos from prior events, make safe online donations, and get reminders or notifications about religious and cultural events. The Visitor's interaction is based on accessibility, ease of use, and involvement. **2. Admin (Member of the Temple Committee):**

Refers to temple officials who have been given permission to make sure that the system's content is correct and complete. Admin users have more rights than regular users. (W3C, 2024) They can upload media assets, control event information, and change content in real time. This makes sure that followers always have access to up-to-date and accurate information. **3. System:**

This is the backend infrastructure that handles important tasks including authentication, data validation, secure file storage, and database synchronization. (Oracle, 2025) The system also automates tasks like making receipts for donations and sending out event reminders, so that users have a smooth experience with all parts of it.

### **Use Cases:**

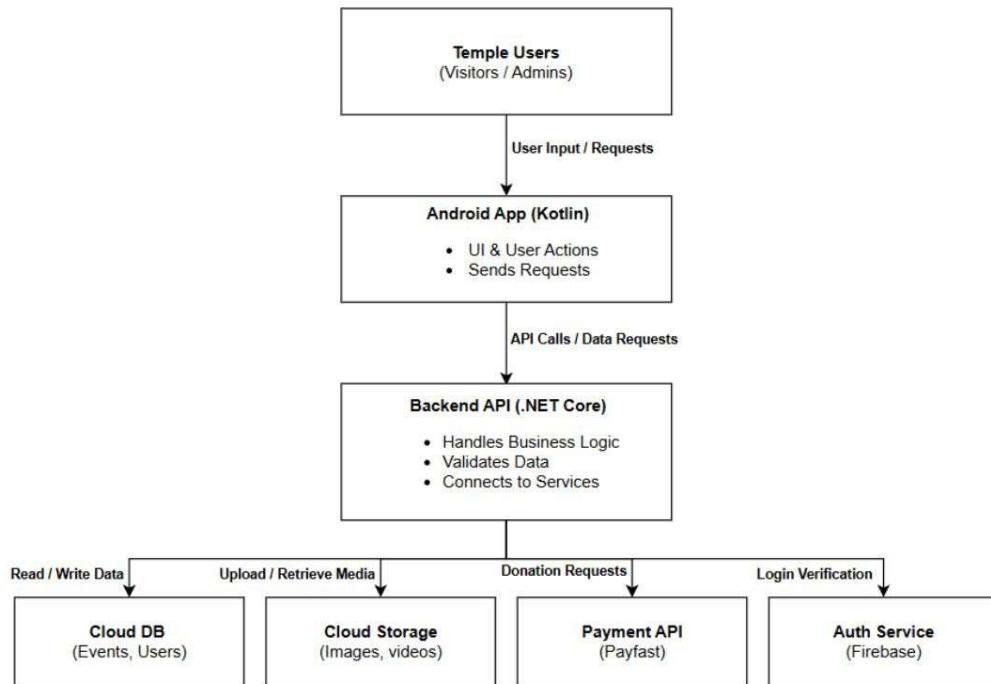
- Register / Login
- View Events

- View Past Event Media
- Donate Online
- Generate Receipt (*included in the donation process*)
- Receive Notifications (*extends from viewing events*)
- Upload Documents and Images (*Admin only*)
- Manage Event Details (*Admin only*)

**Explanation:** The Use Case Diagram shows how different types of users interact with the app to reach their goals. The Visitor (Devotee) can use the public parts of the app, like looking at temple events, watching historical videos, and giving money to temple projects. When a user does something, the backend automatically responds. (W3C, 2024) For example, when someone makes a donation, the system securely processes the payment, updates the donation record, and automatically sends the donor a digital receipt. In the same way, the system can add features by delivering timely notifications to all registered users when a new event is posted. The Admin (Temple Committee Member), on the other hand, works in the app's limited space. Before making any changes, the admin must check in with secure credentials. (W3C, 2024) After verification, they can upload event posters, invitations, pictures, or videos to the temple's cloud storage. This keeps the app up to date with useful information. This regulated access keeps temple data safe and makes sure that everything on the platform stays the same. The System is an important part of the background that makes sure that the app's frontend interface and backend services can talk to each other securely, even though it is not a visible user. (Google Cloud, 2025) It makes sure that all transactions and data activities run smoothly, that storage is reliable, and that authentication is done correctly.

**Purpose:** The objective of this diagram is to show what the Umbilo Temple App can do and how its roles fit together. By mapping out each contact, it makes it clear what the differences are between the rights of regular users and administrators. This picture makes sure that the development team keeps the right access control, data security, and logical system flow. In the end, the diagram gives the app's interaction model a plan, making sure that all of its features fit the temple's goal of using technology to make things more accessible, get people involved in the community, and protect culture.

## **b.) System Architecture Diagram Overview:**



Explanation:

The System Architecture Diagram shows how the Umbilo Temple App's technology is set up and how information flows through it. It also shows how the many parts of the system work together to make sure that both devotees and temple managers have a smooth experience. The design uses a multi-tiered model to make sure that all processes are efficient, scalable, and safe for data.

Temple Users, who are both Visitors (Devotees) and Admins (Temple Committee Members), are at the top of the architecture. The Android mobile app is the main way that these users engage with the system. The app, which was made with Kotlin, lets users do a number of things, like see future temple activities, donate money, and watch videos of past events. The app takes user input and sends it to the backend safely.

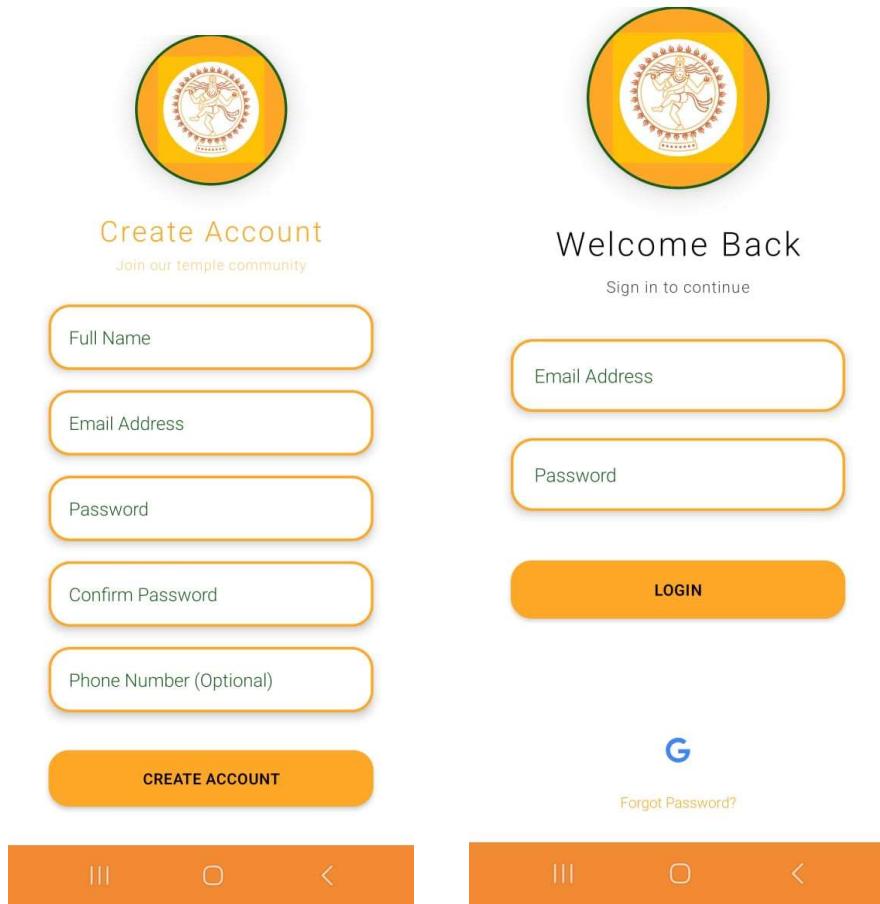
The Backend API, which was made with .NET Core, is the system's main processing unit. It takes care of business logic, checks incoming data, and makes sure that the mobile front end can talk to all of the external services. The API handles all user actions, verifies credentials, keeps track of events, and makes sure that data flows smoothly across the different layers of the system. (Microsoft Learn, 2025)

The cloud-based services that store, manage, and protect the temple's digital resources are below the backend layer. The Cloud Database stores organized information including user profiles, event calendars, and records of donations. (Microsoft Azure, 2025) The Cloud Storage Service is utilized to upload and retrieve multimedia material, including photographs and videos from prior temple events. Integration with the Payment API (PayFast) makes it possible for devotees to make secure online donations through the app. The Authentication Service (Firebase) checks login information to make sure that only authorized users, especially administrators, can utilize management capabilities like making events and uploading media. (Google Cloud, 2025) Unfortunately, we could not add this in for this version.

Each arrow in the diagram represents a specific data flow:

- User Input / Requests: interactions sent from the mobile app.
- API Calls / Data Requests: communication between the app and backend.
- Read / Write Data: data transactions between the backend and database.
- Upload / Retrieve Media: media transfers to and from cloud storage.
- Login Verification: credential authentication via Firebase.

This architecture stresses modularity and integration, which means that each part—frontend, backend, and cloud services—can work on its own but yet work together. The design is flexible enough to add new features in the future, including online integration, real-time notifications, or advanced analytics. This makes it a strong base for the temple's long-term digital strategy.



The Create Account Screen makes it easy and quick for new users to sign up and become part of the temple's online community. It has fields for important user information including full name, email address, password, and confirmation password. (Interaction Design Foundation, 2024) There is also an optional entry for phone number to make communication easier. The UI still has the same calm green and gold color scheme, but the buttons are an orange-gold color that makes things easier to see. The design of the screen makes it easy to read and understand, and it takes users through the steps of creating an account. The software is open to a wide spectrum of users because it lets people sign up by hand or through Google. This makes the registration process more convenient and welcoming.

The Login Screen is the first thing you see when you open the Umbilo Temple App. It is meant to make sure that users can log in safely and easily. For ease, it lets devotees and administrator's login in with their registered email and password or with Google Authentication. The interface has a simple, modern look with a bright white background with green and gold elements that fit with the temple's spiritual style. There is a "Forgot Password?" link to help users get back in, and a "Sign-up" link to help new users sign up. The easy-to-read buttons and basic structure make it easier to use for people of all ages, and the intuitive design and visual hierarchy make it easier for everyone to access.



Welcome, Tehzeeb Govind

Latest from Gallery



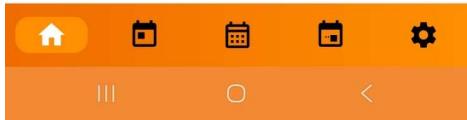
SEE GALLERY

Upcoming Event

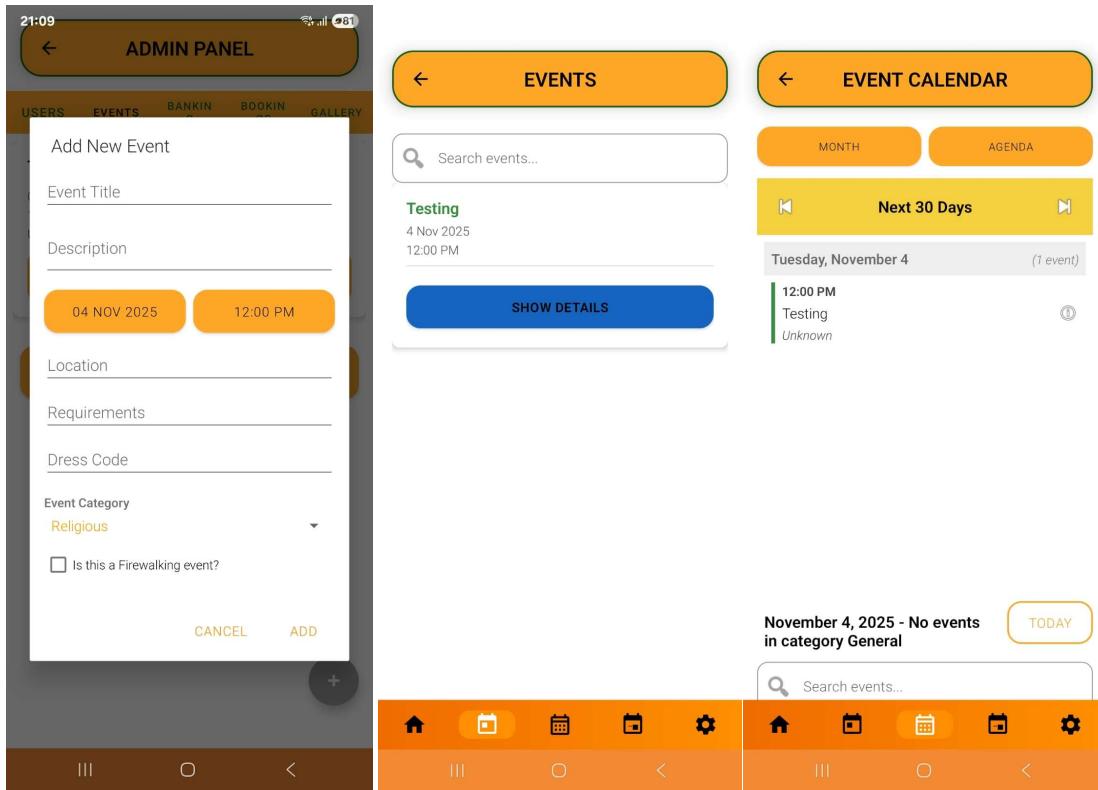
Testing

04 Nov 2025 • 11:34 pm

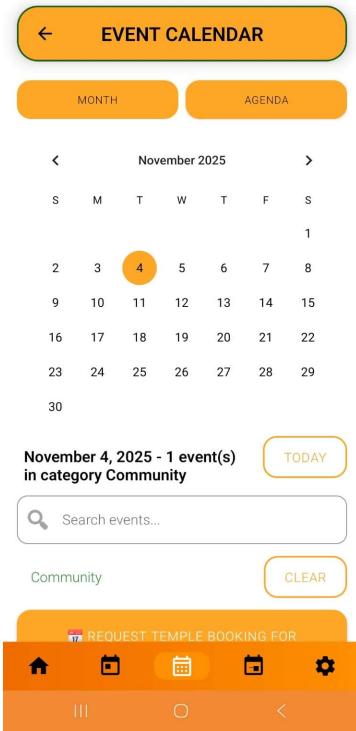
CHECK OUT OUR UPCOMING EVENTS



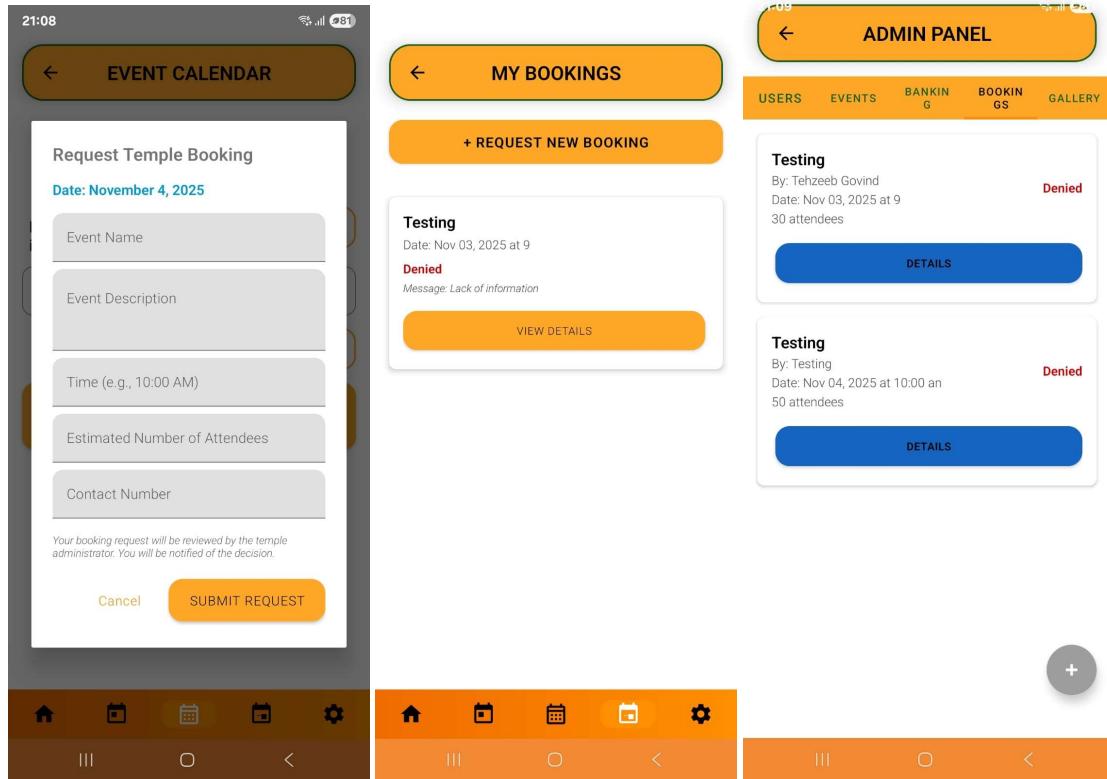
The Home Screen is the main navigation hub. It greets users by name and makes it easy to go to the app's main features. The Events, Donate, and Logout buttons are easy to see in a clean grid structure. The bright yellow and blue icons shine out against the white background (Interaction Design Foundation, 2024) The design is based on user-centered principles, which make sure that all important activities are easy to see at a glance. A bottom navigation bar helps keep things consistent and makes it easy for users to switch between displays. The personalized welcome message lets the user feel more connected to the temple while also making sure that they can easily find their way around and have a good day overall.



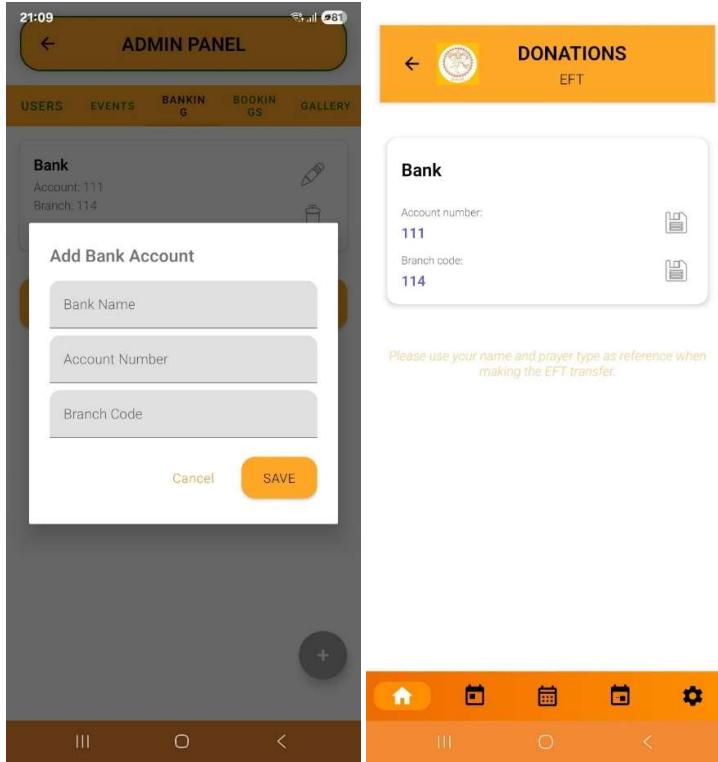
The Events Screen lets users see all the future prayers, temple events, and cultural activities. Its straightforward design makes it easy for consumers to see all the postings in a structured way. Using the same colors for the temple keeps the look constant, while the event icons and calendar integration encourage people to interact. The design encourages participation by giving people the information they need at a glance without making the screen look cluttered. (Google Developers, 2025) This screen's accessibility helps the temple reach its objective of getting people involved and keeping the community involved.



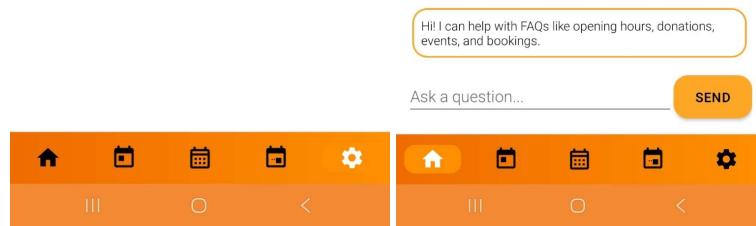
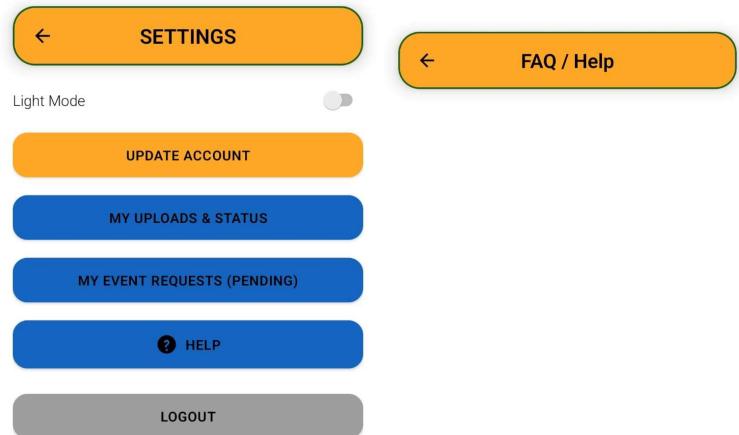
The Event Calendar Screen shows an interactive calendar that lets people look through temple events by date and kind. Users can switch between "Month" and "Agenda" views for more options. Each sort of event is color-coded. There is a "Request Temple Booking" button on the screen that lets devotees reserve specific dates for rituals or prayer sessions right from the app. A "Today" and "Clear" button makes it easy to navigate by letting users go back to today's date or reset their choices. (Nielsen Norman Group, 2024) This feature makes things easier, more organized, and more interesting, which helps devotees schedule their visits to the temple or other spiritual activities.



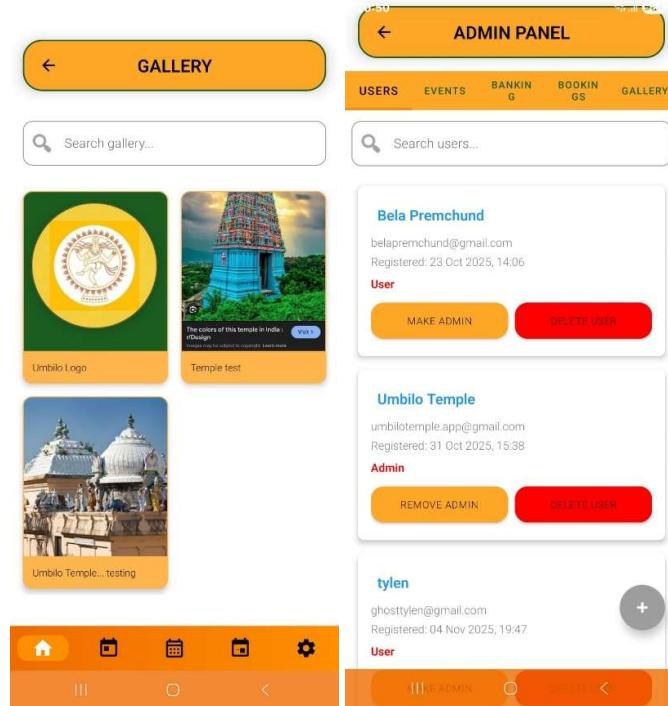
Users may see all of their temple bookings in one place on the My Bookings Screen. It shows a list of past or future appointments, which makes it easy for devotees to keep track of their requests. If there are no bookings, a pleasant message tells customers to hit the "Request New Booking" button to make their first reservation. (Interaction Design Foundation, 2024) The neat structure keeps distractions to a minimum and keeps the brand's white, green and gold colors consistent. The design makes sure that everything is clear and easy to use, so that users can use temple services on their own without needing help from anybody else.



The Donations Screen makes it easy and safe for people to give money to the temple. It shows official bank information, such as the account number, branch code, and the format for donations. By clicking on the clipboard buttons next to each field, users may simply copy this information. This smart feature cuts down on typing mistakes and makes electronic transfers a breeze. At the bottom of the screen, there are directions for how to donate that tell customers to use their name and the sort of prayer they want to make sure the records are correct. (Nielsen Norman Group, 2024) The design shows trust, thankfulness, and simplicity, and it encourages giving by making the contribution process easy and reliable.

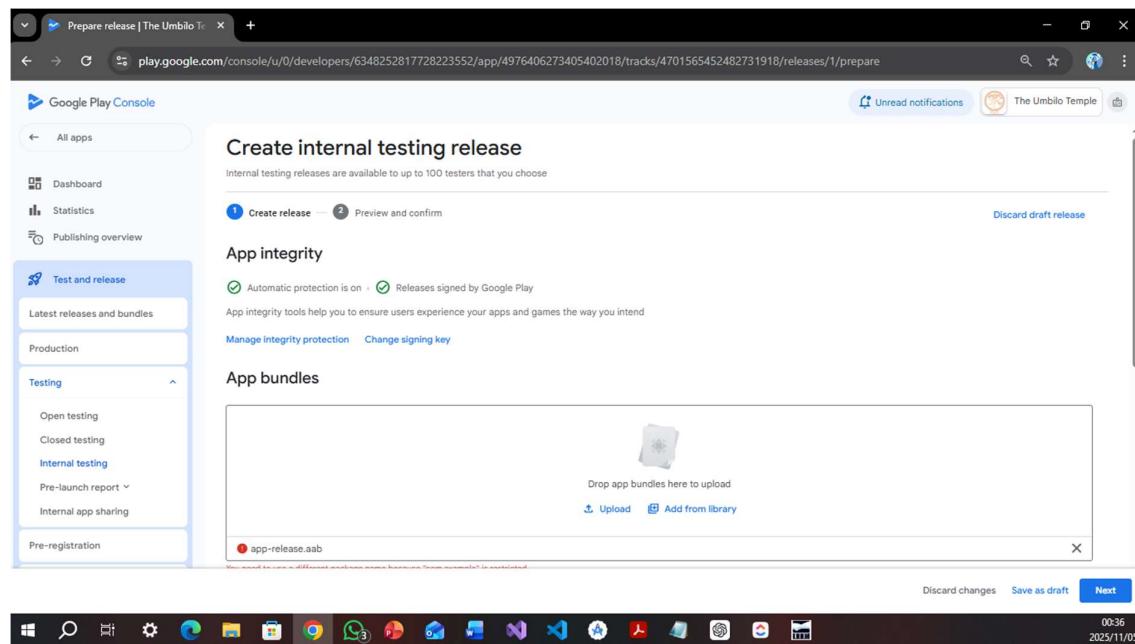


On the settings page, users have the option to use either a dark or light mode for their app. They can also edit their information, view the status of their bookings and even ask for help, which will take them to the chat bot.



Users would upload images which will then be approved by the admin before it is published to the public gallery. The admin users are also only added by other, preset admin users.

## Implementation documents:

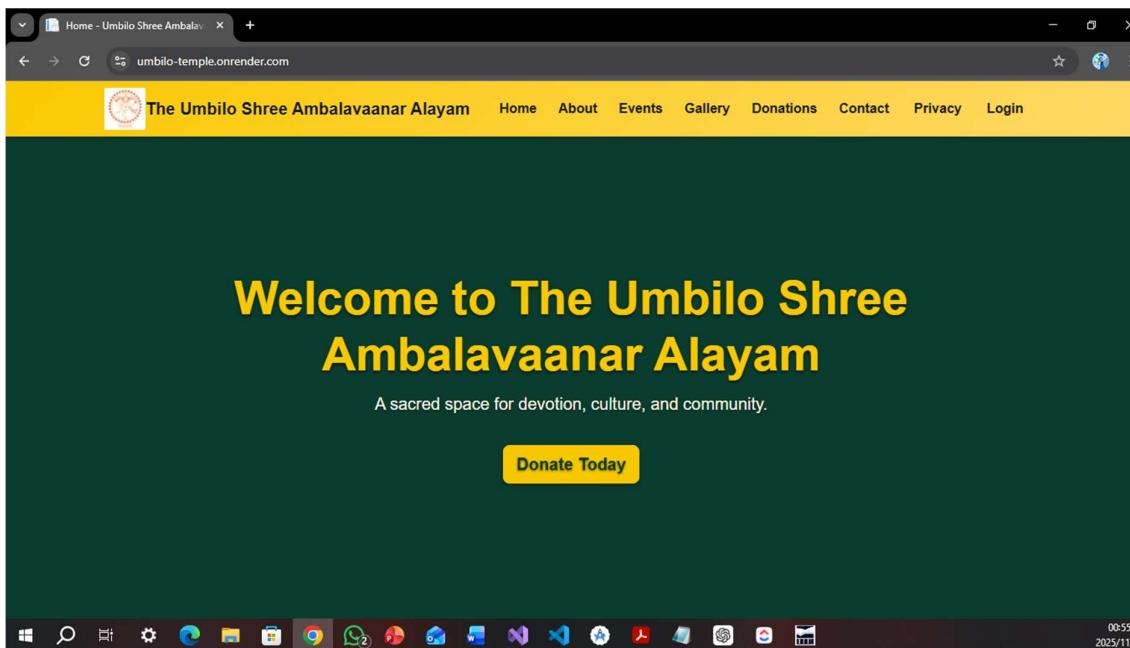


We did experience difficulties when it came to uploading the documents. The error states that they do not take com.example which we did change to com.temple to fix the error. Unfortunately, the bundle still saved it as com.example.

For the website, the UI was created by Kayden and it was then implemented by Altaaf. With this, the design did change once we received the pamphlet from our client. We then decided to implement the colour scheme they user.

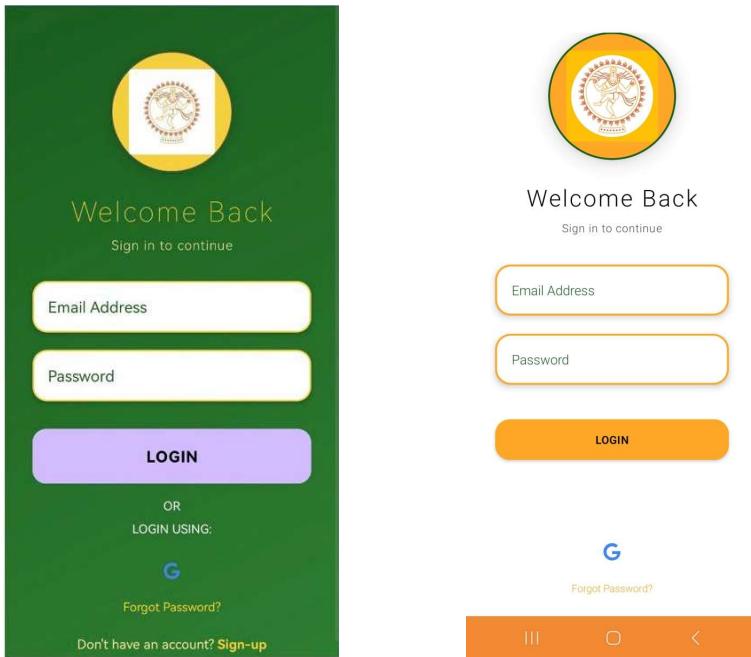


The initial mockup



The current website that we created.

The app UI was created by Altaf. This was then implemented by Mohammed. During the course of this project, with feedback from the team and other external coders, the UI did change from its original green to white, as shown below



For the donations page, the client did not trust the free-tier of 3<sup>rd</sup> party payment system so instead we created a static page where users can see the NPOs banking details and use their own banking apps to make a payment.

The screenshot shows a static page titled "Banking Details" with the following information:

**Bank:** Nedbank

**Account Holder:** UMBILO SHREE AMBALAVAANAR ALAYAM

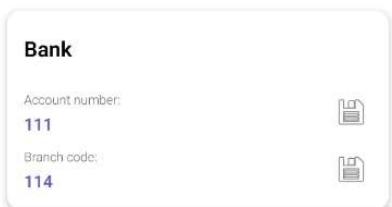
**Account Number:** 1304036316

**Branch Code:** 198765

**Reference:** Name + Prayer Type

A note at the bottom of the box reads: "Please use the reference when making EFT payments."

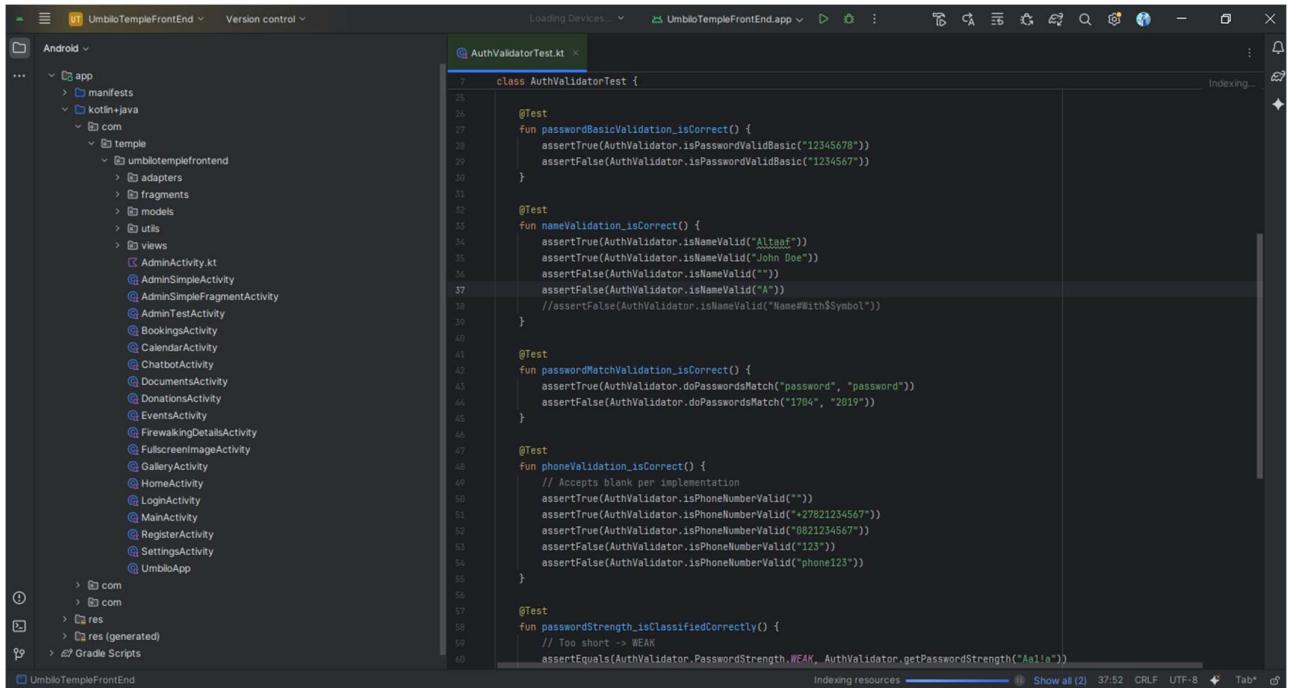
The same applies to the app, admin users can edit this information if needed.



## Testing:

We did ensure to include unit testing to ensure the code works before we commit it to GitHub. This allowed us to see where errors were occurring and how we could fix it.

The image below shows the unit tests for authentication.



A screenshot of the Android Studio interface showing the code editor for a Kotlin test file named `AuthValidatorTest.kt`. The code contains several `@Test` blocks for validating password, name, password match, and phone number. The left sidebar shows the project structure under the `app` folder, including various activities like `LoginActivity`, `MainActivity`, and `SettingsActivity`.

```
class AuthValidatorTest {
    @Test
    fun passwordBasicValidation_isCorrect() {
        assertTrue(AuthValidator.isPasswordValidBasic("12345678"))
        assertFalse(AuthValidator.isPasswordValidBasic("1234567"))
    }

    @Test
    fun nameValidation_isCorrect() {
        assertTrue(AuthValidator.isNameValid("Altaaf"))
        assertTrue(AuthValidator.isNameValid("John Doe"))
        assertFalse(AuthValidator.isNameValid(""))
        assertFalse(AuthValidator.isNameValid("A"))
        //assertFalse(AuthValidator.isNameValid("Name#With$Symbol"))
    }

    @Test
    fun passwordMatchValidation_isCorrect() {
        assertTrue(AuthValidator.doPasswordsMatch("password", "password"))
        assertFalse(AuthValidator.doPasswordsMatch("1704", "2019"))
    }

    @Test
    fun phoneValidation_isCorrect() {
        // Accepts blank per implementation
        assertTrue(AuthValidator.isPhoneNumberValid(""))
        assertTrue(AuthValidator.isPhoneNumberValid("+27821234567"))
        assertTrue(AuthValidator.isPhoneNumberValid("0821234567"))
        assertFalse(AuthValidator.isPhoneNumberValid("123"))
        assertFalse(AuthValidator.isPhoneNumberValid("phone123"))
    }

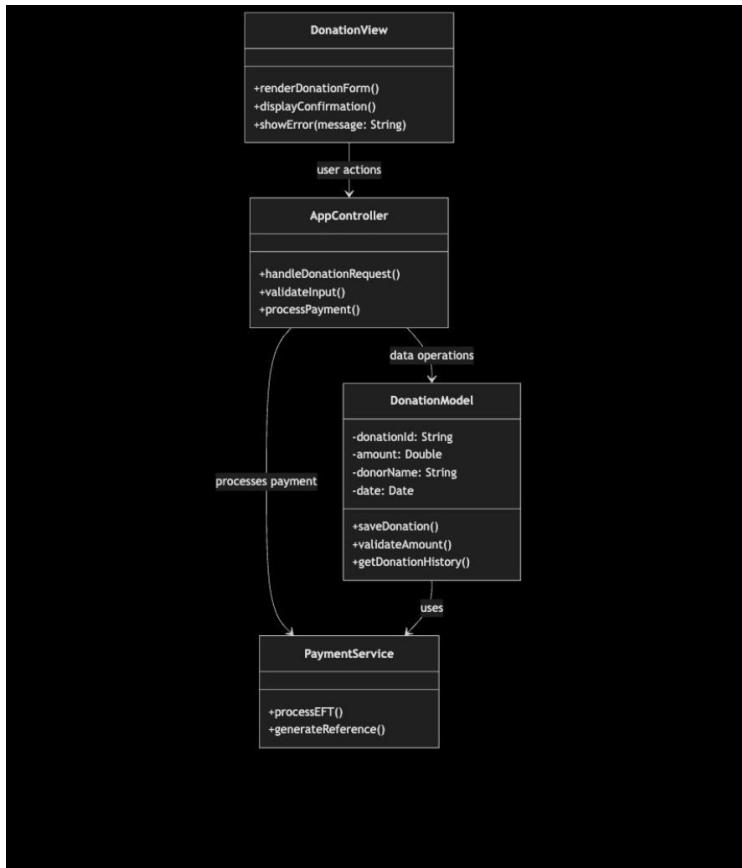
    @Test
    fun passwordStrength_isClassifiedCorrectly() {
        // Too short -> WEAK
        assertEquals(AuthValidator>PasswordStrength.WEAK, AuthValidator.getPasswordStrength("Aa1l@"))
    }
}
```

## Architecture artifacts:

### Design Patterns

The Umbilo Temple Web App was designed with a focus on simplicity, reliability, and long-term maintainability. Since the app is mainly used by older temple members, the system's structure had to be stable, secure, and easy to expand later. To achieve that, a few key software design patterns were applied during the design phase. These patterns help keep the system organized and make it easier to add new features without breaking existing ones. (Freeman, et al., 2004; Buschmann, et al., 1996)

## 1. Model–View–Controller (MVC) Pattern



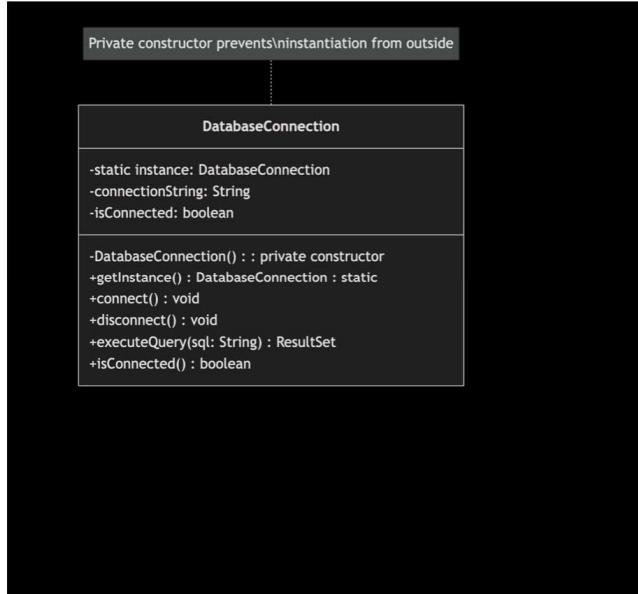
The MVC pattern divides the system into three main parts:

- Model – deals with the data (such as users, events, and donations).
- View – handles what users see on screen, like the donation page or events calendar.
- Controller - acts as the link between the user interface and the data, controlling how the app responds to user input.

In the Umbilo Temple Web App, this approach keeps the code clean and well separated. For example, if the temple changes how events are displayed, developers can update the view without touching the logic behind it. (Freeman, et al., 2004)

## 2. Singleton Pattern

### Singleton Pattern

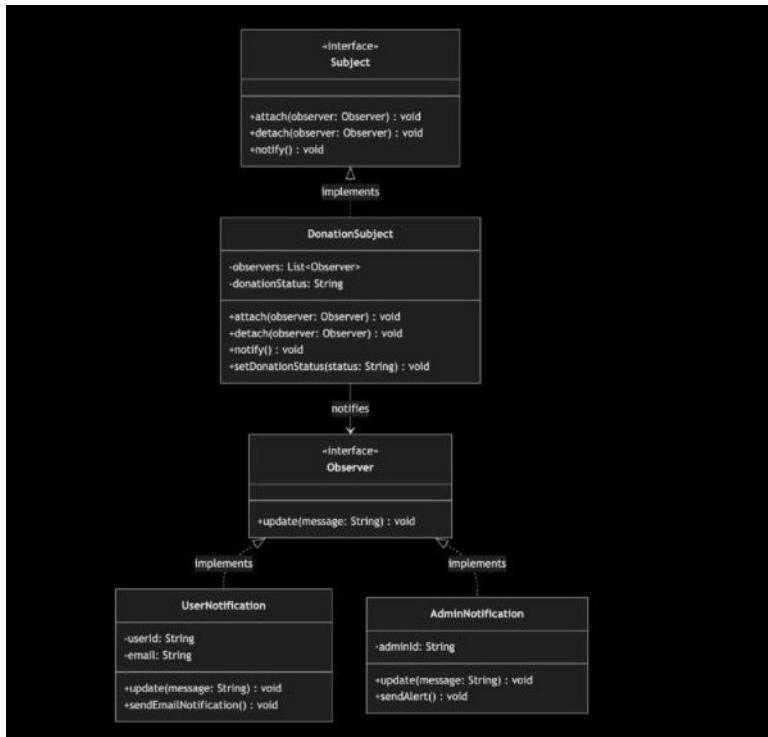


The Singleton pattern makes sure that there's only one instance of a specific class in the system.

In this project, it's used for the Database Connection Manager, which controls how the app connects to the database. Having just one connection active at a time prevents errors, reduces load, and helps maintain data accuracy.

This is especially important when multiple users log in or make donations at the same time. (Buschmann, et al., 1996; Gamma, et al., 1994)

### 3. Observer Pattern

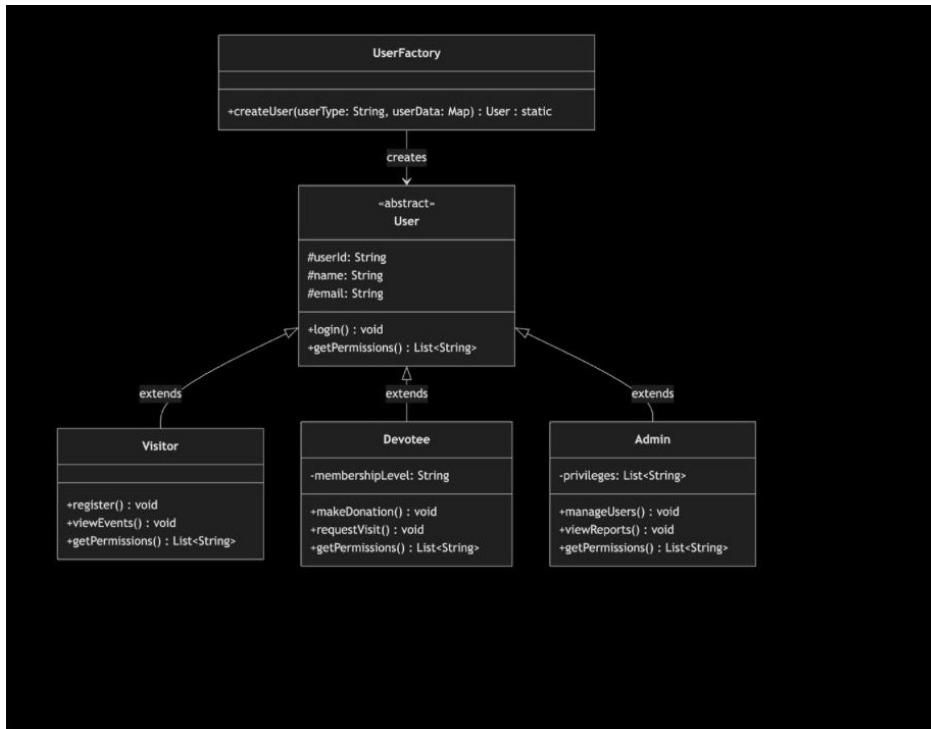


The Observer pattern allows one part of the system to automatically update other parts when something changes.

In the Umbilo Temple App, this can be used for sending notifications for example, when an admin adds a new event, all subscribed users could automatically get a message or email about it.

This keeps temple members informed without requiring them to manually check the app every day. (Freeman, et al., 2004)

#### 4. Factory Pattern



The Factory pattern is used to create different types of objects without having to know exactly which class is being created.

In the app, this pattern is used when creating different user roles, such as Visitor, Donor, or Admin. The factory decides which user type to create depending on what information is provided during registration.

It makes the system more flexible and easier to expand later for example, if a new “Volunteer” role is added, it can be created through the same factory. (Freeman, et al., 2004)

#### Opportunities for Improvement

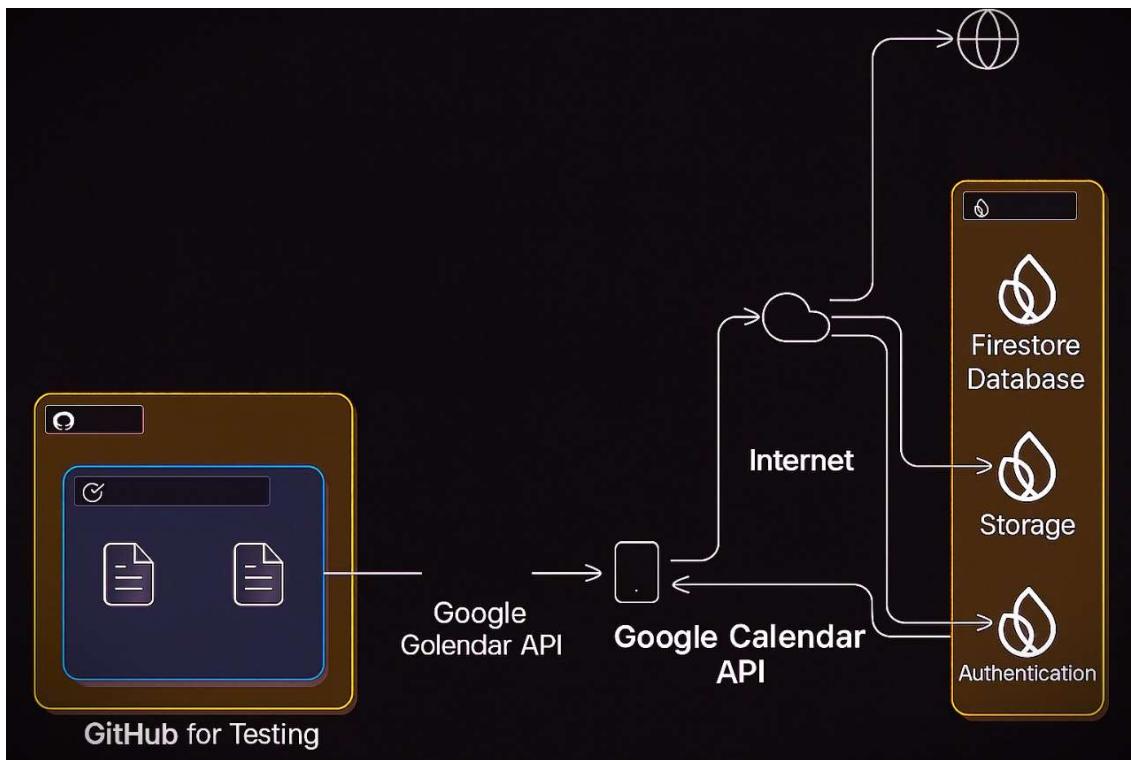
Although the current design is effective, there's still room for improvement:

- The Observer pattern could be extended to support push notifications for instant updates.
- The Factory pattern could include built-in permissions for better user control.
- The MVC structure could be implemented using a framework like Django or Laravel to further streamline development.

Overall, these patterns keep the Umbilo Temple Web App well structured and easy to maintain. They make the system easier to understand, prevent duplicate work, and ensure that new features like notifications or additional user roles can be added smoothly in the future.

## Cloud Architecture

The cloud architecture of the Umbilo Temple mobile app makes use of Google Play Store for material that is visible to the public, Firebase Cloud Storage for media, Firebase Authentication for user management, and Firebase Realtime Database for data storage. (Preston, 2025)



### Firebase Realtime Database:

- Purpose: Stores structured app data, such as information on events, donations, and gallery content. Realtime Database was chosen for its real-time data syncing capability, which allows instant updates across all devices connected to the app.
- Network Segregation and Protocol: Realtime Database communicates over HTTPS to ensure secure data transmission between the client app and the database. This provides fast, real-time updates for app users, making it especially suitable for displaying events and gallery updates. (Firebase, n.d.)

#### Firebase Authentication:

- Purpose: Manages administrator logins with a secure, password-based system. Only administrators can log in to access restricted app management features, while general users have browsing-only access. (Firebase, n.d.)
- Security Protocol: Authentication requests are encrypted over HTTPS, and Firebase's role-based access control is used to restrict administrative functions to logged-in admins only. (Firebase, n.d.)

#### Firebase Cloud Storage:

- Purpose: Used to store media files like images and videos for the gallery.

Firebase Cloud Storage provides scalable storage and secure access to files, allowing administrators to manage content while general users have read-only access.

- Access Control: Only admins have permissions to upload content, while public users can view images. Firebase's security rules ensure access is restricted accordingly. (Firebase, n.d.)

## Security:

#### Network Security:

- HTTPS Encryption: The app, Firebase services, and Google Play Store all transmit data via HTTPS. This encryption safeguards sensitive information, including personal data and logon credentials, from interception during transmission.
- Firebase Realtime Database Rules: Custom security policies are implemented in Firebase Realtime Database to regulate data access. These regulations enforce read and write restrictions that are determined by user roles, guaranteeing that only authorised administrators have the ability to manage sensitive content.
- Network Segregation: We can improve our control over various categories of network traffic by segregating Google Play Store for public-facing content and Firebase-based services (data, authentication, and storage). This approach limits the attack surface and exposure. (Android Developers, n.d.)

#### Authentication and Access Control:

- Firebase Authentication: Administrator access is secured through Firebase Authentication with role-based access controls. Only administrators can log in and access sensitive management features, while all other users can browse the app without login requirements.

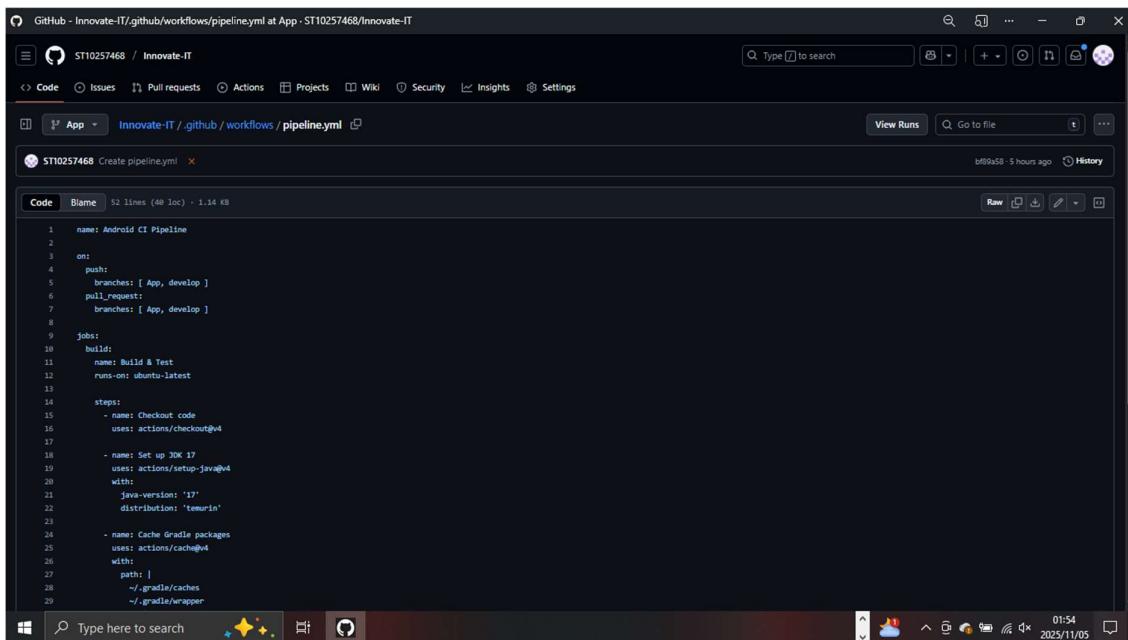
- Password Security: Firebase Authentication uses secure, salted, and hashed passwords, and enforces secure password policies to prevent unauthorised access. (Android Developers, n.d.)

### Cloud Storage Security:

- Firebase Cloud Storage Rules: Firebase Cloud Storage security rules ensure that only administrators can upload media, while all users have read-only access to view content in the gallery. This setup prevents unauthorised modifications or deletions of media files.
- Access Control for Media: Media files are protected using Firebase's security rules, which prevent unauthorised downloads or access to restricted content, minimising data exposure. (Android Developers, n.d.)

### DevOps:

Our pipeline checks and ensures the code is functional and safe before it can be deployed or committed. With the use of Security secrets on GitHub, we were also able to protect our API keys and secret information. (ChatGPT, 2025)



```

name: Android CI Pipeline

on:
  push:
    branches: [ App, develop ]
  pull_request:
    branches: [ App, develop ]

jobs:
  build:
    name: Build & Test
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v4
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Cache Gradle packages
        uses: actions/cache@v4
        with:
          path: |
            -./gradle/caches
            -./gradle/wrapper

```

## Running Costs:

For the duration of this project, our NPO was not charged to use any platform.

For the website hosting and database, we used free tiers. Since the NPO is in an unstable state financially as of current, they could not afford to use a paid third party payment platform, paid tiers in Firebase, or a personal domain.

For the shirt and prize for the raffle, we used the money that we received as a sponsorship amounting to R 2500 (proof of sponsorship is attached at the bottom of the document).

The money was spent as follows:

1. Shirts Purchased from Makro – R89 x 8 = R 712
2. Print – R 120 x 8 = R960
3. Logo redesign – R250
4. The rest went to the hamper for various gifts (purchased on sale)

We project that by June 2026, our NPO should be in a better position financially. We would then be able to integrate new features such as a functional donations page, an all-access database, and a personal domain.

We do plan to use:

- Firebase for the database, which would cost R 70/month (Firebase, n.d.)
- PayFast for NPOs which would be 3.5% estimated per transaction, this number was obtained when we contacted PayFast enquiring about their NPO discount.

## Change Management:

1. Donations page: we planned on including a full functional donations page, but unfortunately our NPO cannot afford to pay for a third party pay wall so we changed our design to be a static page. The second option was to integrate a GoFundMe page, but it did not allow for an account to be created for an NPO in South Africa after setting the region to South Africa.
2. The main issue was that the public was not aware of this historic site, hence we created the website in hopes that it would reach more people nation wide. During the comrades, it was stated in a pamphlet, given to international runners, that The Umbilo Temple is a ‘Must See’ site.
3. The committee members and the Guru at the temple had an issue with management and timetables. With the use of the app, we ensured for improved management of events, images, videos and prayer appointments. We used a

- colour coded system to differentiate between the different types of appointments, whether it is for a prayer, a visit or a community event, it can be easily seen.
4. 50% of the team do plan on maintaining and updating the app and the website as the years go on, pro bono.

## Additional resources:

1. NDA and MOA from client  
It is added to the submission with this file.
2. The team's declarations:

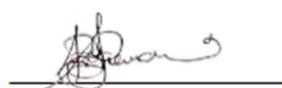
### Declaration of authenticity

I, Bela Jyoti Premchund ID Number, 0504300181089

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



**STUDENT SIGNATURE**

03 November 2025

**DATE**

## Declaration of authenticity

I, Kayden Padayachee ID Number, 0403295352088

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



---

---

03 November 2025

STUDENT SIGNATURE

DATE

## Declaration of authenticity

I, Altaf ally, ID Number, 0402175680089

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



STUDENT SIGNATURE

DATE 04/11/25

## Declaration of authenticity

I, Mohammed Tehzeeb Govind ID Number, 0405265678087

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



STUDENT SIGNATURE

DATE **04/11/25**

## Declaration of authenticity

I, **Sohum Singh** ID Number, **0405265886086**

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



STUDENT SIGNATURE

DATE **04/01/25**

---

Declaration of authenticity

I, Kimera pillay ID Number, 0412120125080

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.

Kimera Pillay

STUDENT SIGNATURE

4/11/2025

DATE

## Declaration of authenticity

I, Kayla Shunmugam ID Number, 030711192083

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



STUDENT SIGNATURE

DATE 4 November 2025

## Declaration of authenticity

I, Imaad kajee ID Number, 0306295066089

hereby declare that this portfolio, and any evidence included therein, contains my own independent work and that I have not received help from other groups.

I confirm that we have not committed plagiarism in the accomplishment of this work, nor have I falsified and/or invented experimental data.

I accept the academic penalties that may be imposed for violations of the above.



STUDENT SIGNATURE

DATE 4 November  
2025

### 3. ClickUp Screenshots:

The image displays two side-by-side screenshots of the ClickUp application interface, showing different project boards.

**Top Screenshot (Documentation Project):**

- Project:** Documentation
- Task Statuses:**
  - TO DO: 0
  - IN PROGRESS: 0
  - IN REVIEW: 1
  - COMPLETE: 17
- Tasks:**
  - User Story (Due Jul 18, 12:00 pm)
  - Presentation Mock-Ups (Due Jun 18, 12:00 pm)

**Bottom Screenshot (Development - App Project):**

- Project:** Development - App
- Task Statuses:**
  - TO DO: 1
  - IN PROGRESS: 1
  - IN REVIEW: 4
  - COMPLETE: 4
- Tasks:**
  - Group Demo (Due Oct 17, 10:00 am)
  - Donations Page (Due Jul 25, 5:00 pm)
  - App UI
  - App Authentication
  - Events
  - Booking Calendar

The image displays two side-by-side screenshots of a task management application's interface, titled "Innovate IT".

**Screenshot 1: Development - Website | Innovate IT (Board)**

This screenshot shows the "Development - Website" board. The sidebar on the left lists "Home", "Planner", "Brain", "Teams", "Docs", "Dashboard", "Whiteboard", "More", "Invite", and "Upgrade". Under "Spaces", it shows "All Tasks - Innovate IT" and "Team Space". In "Team Space", there are sections for "Projects" (Documentation, Development - App, Development - We...), "Documentation", "Presentation", "Testing", and "New Space". Under "Channels", there is a "Development - App" channel.

The main area shows a Kanban board with columns: TO DO (1 card, "Events Page"), IN PROGRESS (0 cards), IN REVIEW (0 cards), and COMPLETE (5 cards, including "UI", "Donation Page", "Home Page", "About Page", and "Security Doc"). A "1 subtask" button is visible under the TO DO column. Buttons for "+ Add Task" are located at the bottom of each column.

**Screenshot 2: Presentation | Innovate IT (Board)**

This screenshot shows the "Presentation" board. The sidebar and space structure are identical to the first screenshot. The main area shows a Kanban board with columns: TO DO (1 card, "Business cards"), IN PROGRESS (0 cards), IN REVIEW (0 cards), and COMPLETE (2 cards, including "Shirts" and "Photos"). A "Save" button is visible on the "Business cards" card. Buttons for "+ Add Task" are located at the bottom of each column.

**Planning | Innovate IT**

File Edit View History Window

Innovate IT

Dashboards / Planning

Edit mode

0 Unassigned Tasks    0 tasks in progress    27 tasks completed

Refreshed just now Auto refresh: On Schedule report + Card

**Total Tasks by Assignee**

| Assignee                   | Percentage |
|----------------------------|------------|
| Unassigned                 | 5.88%      |
| Kimera Pillay              | 9.8%       |
| Tehzeeb Govind             | 23.52%     |
| altaaf                     | 13.72%     |
| Kayden Padayachee          | 11.76%     |
| st100504086@vconnect.edu.z | 7.8%       |

**Open Tasks by Assignee**

| Assignee       | Tasks |
|----------------|-------|
| altaaf         | 3     |
| Tehzeeb Govind | 2     |

Tasks

- 3 Unassigned
- 5 altaaf
- 6 BP Bela Jyoti Premchund
- 4 OP Kayden Padayachee
- 4 KK Kayla Kimberly
- 5 VP Kimera Pillay
- 2 SS Suhum Singh
- 1 st100504086@vconnect.edu.z

Calculate

**Latest Activity**

**Development | Innovate IT**

File Edit View History Window

Innovate IT

Dashboards / Development

Edit mode

Unassigned Tasks    tasks in progress    tasks completed

Refreshed Apr 27 Auto refresh: On Schedule report + Card

**Total Tasks by Assignee**

| Assignee             | Percentage |
|----------------------|------------|
| Tehzeeb Govind       | 43.75%     |
| altaaf               | 25%        |
| Bela Jyoti Premchund | 12.5%      |
| Suhum Singh          | 9.37%      |

**Open Tasks by Assignee**

| Assignee       | Tasks |
|----------------|-------|
| altaaf         | 5     |
| Tehzeeb Govind | 8     |

Tasks

- 2 Unassigned
- 1 altaaf
- 2 BP Bela Jyoti Premchund
- 2 SS Suhum Singh
- 1 TG Tehzeeb Govind

Calculate

**Latest Activity**

Today

- Security Doc

4. Sponsorship:

## **KEVDEN PANEL & RECOVERIES PTY LTD & HighBar Electrical & Maintenance PTY LTD**

Company Reg No: 2019/434343/07  
VAT No: 4760263444  
kevdenptld@gmail.com

Company Reg No: 2019/628736/07  
VAT: 4680304542

**Sponsorship form:**

Please complete all of the fields below in full to ensure that your sponsorship request for the IT Assessment Group at Varsity College can be processed without delay. This sponsorship has been requested on behalf of the group by our employee, Miss Kimera Pillay, and will be generously provided by Kevden Panel and Recoveries PTY LTD and Highbar Electrical and maintenance PTY LTD. Once you have entered every required detail, your application will be reviewed and sponsored funds or resources allocated accordingly.

**Sponsored party details:**

|                                |   |
|--------------------------------|---|
| <b>Full Name:</b>              | Bela Jyoti Premchund  |
| <b>ID Number:</b>              | 0504300181089   |
| <b>Institution:</b>            | IIE Varsity College Westville   |
| <b>Reason for sponsorship:</b> | As a group of final year IT student in Varsity College, we are required to create an app for an NPO and we will be presenting our app to other IT companies. I am asking for sponsorship for our company, Innovate IT, to get our shirts and business cards printed for our presentation. |
| <b>Amount Requested (ZAR):</b> | R 2500  |

## References

- Android Developers.** (n.d.) *Improve your app's security.* [Online]  
Available at: <https://developer.android.com/privacy-and-security/security-best-practices>  
[Accessed 3 November 2025].
- AWS.** (2025) *Cloud Security Overview.* [Online]  
Available at: <https://aws.amazon.com/security/>  
[Accessed 21 October 2025].
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M.** (1996) *Pattern-oriented software architecture: a system of patterns.* Chichester: John Wiley & Sons.  
[Online]  
Available at: <https://www.wiley.com/en-us/Pattern+Oriented+Software+Architecture%2C+A+System+of+Patterns-p-9780471958697>  
[Accessed 31 October 2025].
- ChatGPT.** (2025) *Fix gradlew path error.* [Online]  
Available at: <https://chatgpt.com/share/690a9494-73ec-8006-b3fd-304d7585a232>  
[Accessed 22 October 2025].
- ChatGPT.** (2025) *WIL Project.* [Online]  
Available at: <https://chatgpt.com/g/g-p-68ffc8dd1f448191a66690a9e6f91ecd-wil/project>  
[Accessed 27 October 2025].
- Firebase.** (n.d.) *FCM Architectural Overview.* [Online]  
Available at: <https://firebase.google.com/docs/cloud-messaging/fcm-architecture>  
[Accessed 1 November 2025].
- Freeman, E., Robson, E., Bates, B. and Sierra, K.** (2004) *Head First Design Patterns.*  
[Online]  
Available at: <https://www.oreilly.com/library/view/head-first-design/0596007124/>  
[Accessed 31 October 2025].
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J.** (1994) *Design Patterns: Elements of Reusable Object-Oriented Software.* [Online]  
Available at: <https://dl.acm.org/doi/book/10.5555/186897>  
[Accessed 31 October 2025].
- GeeksforGeeks.** (2025) *Architectural Design – Software Engineering.* [Online]  
Available at: <https://www.geeksforgeeks.org/software-engineering-architectural->

[design/](#)

[Accessed 1 November 2025].

**GeeksforGeeks.** (2025) *Difference between Software Design and Software Architecture.* [Online]

Available at: <https://www.geeksforgeeks.org/difference-between-software-design-and-software-architecture/>

[Accessed 18 October 2025].

**GeeksforGeeks.** (2025) *Enhanced ER Model.* [Online]

Available at: <https://www.geeksforgeeks.org/dbms/enhanced-er-model/>

[Accessed 30 September 2025].

**GeeksforGeeks.** (2025) *Fundamentals of Software Architecture.* [Online]

Available at: <https://www.geeksforgeeks.org/software-engineering/fundamentals-of-software-architecture/>

[Accessed 24 October 2025].

**GeeksforGeeks.** (2025) *How to Become a Software Architect.* [Online]

Available at: <https://www.geeksforgeeks.org/how-to-become-a-software-architect/>

[Accessed 20 October 2025].

**GeeksforGeeks.** (2025) *How to Draw Entity Relationship Diagrams (ERDs).* [Online]

Available at: <https://www.geeksforgeeks.org/sql/how-to-draw-entity-relationship-diagrams/>

[Accessed 16 October 2025].

**GeeksforGeeks.** (2025) *Introduction of ER Model.* [Online]

Available at: <https://www.geeksforgeeks.org/dbms/introduction-of-er-model/>

[Accessed 14 October 2025].

**GeeksforGeeks.** (2025) *Relationships in ER Model.* [Online]

Available at: <https://www.geeksforgeeks.org/dbms/relationships-in-er-model/>

[Accessed 19 October 2025].

**GeeksforGeeks.** (2025) *Difference between Functional and Non-Functional Requirements.* [Online]

Available at: <https://www.geeksforgeeks.org/difference-between-functional-and-non-functional-requirements/>

[Accessed 18 August 2025].

**Google Cloud.** (2025) *Firebase Authentication Documentation.* [Online]

Available at: <https://firebase.google.com/docs/auth>

[Accessed 21 October 2025].

- Google Developers.** (2025) *Material Design Guidelines for Android Apps*. [Online]  
Available at: <https://m3.material.io/>  
[Accessed 24 October 2025].
- Harvard Business School.** (2023) *Database Best Practices*. [Online]  
Available at: <https://www.hbs.edu/research-computing-services/data-practices/database-best-practices/>  
[Accessed 27 October 2025].
- Harvard University Information Technology (HUIT).** (2024) *Digital Accessibility Policy*.  
[Online]  
Available at: <https://accessibility.huit.harvard.edu>  
[Accessed 20 October 2025].
- IBM.** (2025) *Cloud Architecture Design Principles*. [Online]  
Available at: <https://www.ibm.com/cloud/architecture>  
[Accessed 26 October 2025].
- Interaction Design Foundation.** (2024) *What is UX Design?* [Online]  
Available at: <https://www.interaction-design.org/literature/topics/ux-design>  
[Accessed 18 September 2025].
- Microsoft Azure.** (2025) *Introduction to Azure Blob Storage*. [Online]  
Available at: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>  
[Accessed 18 October 2025].
- Microsoft Learn.** (2025) *ASP.NET Core MVC Overview*. [Online]  
Available at: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview>  
[Accessed 19 October 2025].
- Nielsen Norman Group.** (2024) *10 Usability Heuristics for User Interface Design*.  
[Online]  
Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/>  
[Accessed 21 October 2025].
- Oracle.** (2025) *Database Design and ERD Basics*. [Online]  
Available at: <https://docs.oracle.com/en/database/>  
[Accessed 30 October 2025].
- Preston, R.** (2025) *What Is Cloud Architecture? (With Benefits, Types and Tips)*. [Online]  
Available at: <https://www.indeed.com/career-advice/career-development/cloud-architecture>  
[Accessed 1 November 2025].

**TechTarget.** (2025) *What Are Functional and Non-Functional Requirements?* [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/functional-vs-nonfunctional-requirements> [Accessed 15 August 2025].

**W3C.** (2024) *Web Content Accessibility Guidelines (WCAG) 2.2.* [Online] Available at: <https://www.w3.org/WAI/standards-guidelines/wcag/> [Accessed 20 October 2025].

## AI Declaration:

We did use AI in the project.

The reasons are as follows:

- Fixing and getting the colour codes for the website
- Fixing the pipeline errors
- Ordering all references collected by everyone
- Determining if SonarQube or Snyk would be better – we still used both as each had their own advantages
- For uploading the app to Google Play, some of the images needed to be resized, ChatGPT helped us with finding the correct way and website to do so
- We needed ChatGPT to fix the buttons and the privacy policy on the website as the buttons were there but they were not functioning.
- We also used ChatGPT to assist us in dissecting the POE and helping us find the right hosting platform.

The links are as follows:

ChatGPT, 2025. *Fix gradlew path error.* [Online] Available at: <https://chatgpt.com/share/690a9494-73ec-8006-b3fd-304d7585a232> [Accessed 22 October 2025].

ChatGPT, 2025. *WIL.* [Online] Available at: <https://chatgpt.com/g/g-p-68ffc8dd1f448191a66690a9e6f91ecd-wil/project> [Accessed 27 October 2025]. – this link is for the folder that has all the files and different chats we used for this project.