**Project Requirements Document (PRD)**

|  |  |
| --- | --- |
| **Section** | **Details** |
| 1. Overview | The CAMPUS NAVIGATION AND RESOURCE FINDER SYSTEM is a digital application aimed at enhancing the campus experience for students, staffs, and visitors by providing an intuitive platform for navigating and locating various campus resources. The app leverages modern technologies to offer an interactive map, real-time navigation, and location-based services |
| Project Title | NAVIGATION AND RESOURCE FINDER APPLICATION |
| Document Version | 1.0 |
| Date | [Date of the document] |
| Prepared by | MBINJI NGANGA |
| 2. Executive Summary |  |
| Project Purpose | Easy access to learning resources and materials |
| Objectives | * **Improve campus navigation:** proved users with turn-by-turn navigation to classrooms, offices, cafeterias, and other facilities. * Increase accessibility: include accessible routes and options for users with disability. * Resource Discovery: helps uses to find resources like classrooms, study areas, parking, and even locations. |
| Scope | * Interactive campus map * Turn-by-Turn Navigation * Search functionality |
| 3. Project Background | * The navigation and resource finder app will be first navigation solution specifically developed for the campus. It aims to solve challenges in navigating the campus, finding resources, and accommodating accessibility needs with an interactive, digital map. * By using a modern stack like react.js, python with Django and PostgreSQL with GIS hosted in google cloud, the app offers a user-friendly experience. This first-of-its-kind solution will set the foundation for future upgrades, such as advanced indoor positioning or augmented reality navigation. |
| Business Need | The business need for the campus navigation and resource finder app arises from the increasing complexity of the university, which often includes numerous building, facilities, and resources spread across large areas. |
|  |
| Target Audience |  **Students/Learners**: New and existing students who need help to navigate the school to find classrooms, libraries, study areas, and event location. It will be especially useful for new students who are unfamiliar with the campus layout.   **University staff and faculty: staff members and faculty who needs to locate offices, meeting rooms, conference halls, or other facilities.**   **Campus visitors: individuals visiting the campus for events, tours, or conference will benefit from real-time navigation and information about amenities, parking and point of interest.** |
| Stakeholders | **Primary stakeholder**   * Students * University stall and faculty * Campus visitors   **Secondary stakeholders**   * University administration * IT department * Accessibility services |
| 4. Product Features | 1. Interactive map 2. Search and filter options 3. Resource discovery 4. User authentication and personalization 5. Backend and API Integration |
| Core Features | * Interactive campus map * Real-time Navigation * Turn-By-Turn Navigation * Search and Resource Finder * User Authentication and Personalization |
|  |
| Additional Features | * Offline map access * Location sharing * Customizable map layers |
|  |
| User Roles and Permissions |  **Students**   * **Permissions**:   + Access all standard app features, including searching for locations, navigation, and event information.   + Log in to personalize the experience (e.g., save favorite locations, view recent searches).   + Receive push notifications for campus updates or event reminders. * **Limitations**:   + Cannot access admin-specific features or make permanent changes to map data.    **Faculty and Staff**   * **Permissions**:   + Similar to student permissions, with the additional ability to view internal points of interest, such as staff-only areas.   + May have limited editing capabilities for updating information related to their departments, such as events or office hours. * **Limitations**:   + Cannot make significant changes to the map structure or manage other users.    **Campus Visitors**   * **Permissions**:   + Access the app's general navigation and search features without logging in.   + View public event information and locations. * **Limitations**:   + No personalized features or access to restricted locations.   + Limited or no ability to save preferences or receive notifications.    **Administrators**   * **Permissions**:   + Full access to all app features, including managing user roles and permissions.   + Ability to update map data, verify crowdsourced updates, and manage events.   + Access to analytics dashboards for monitoring app usage and generating reports. * **Limitations**:   + Should adhere to data security protocols when handling user data or sensitive information.    **IT/Technical Support**   * **Permissions**:   + Manage backend infrastructure, including database maintenance, API management, and system updates.   + Access to admin tools for troubleshooting and technical support. * **Limitations**:   + May not have access to administrative features unrelated to technical support. |
|  |
| 5. Functional Requirements |  **User Authentication and Authorization**   * The app must allow users to sign up, log in, and log out using OAuth 2.0 for secure authentication. * User roles (students, faculty, visitors, administrators) must be defined with appropriate permissions. * Personalized user settings, such as saved locations and navigation preferences, should be accessible after login.    **Interactive Map Interface**   * The app must display an interactive map of the campus that allows users to pan, zoom, and click on different points of interest (POIs) for more information. * The map should include features like buildings, pathways, restrooms, parking areas, and other significant locations. * Users must be able to filter map results based on categories such as academic buildings, dining areas, or accessible routes.    **Navigation and Route Guidance**   * The app must provide turn-by-turn navigation, guiding users from their current location to the selected destination on the map. * Route options should include preferences for shortest path, accessible routes, and user-selected POIs. * Real-time location updates should be available to show progress along the chosen route.    **Search Functionality**   * Users must be able to search for locations on the campus, including specific buildings, rooms, and facilities. * The search results should include relevant details, such as location descriptions, hours of operation, and contact information if applicable. * Category-based filtering options should be available in search results for quick access to specific types of locations (e.g., dining halls, restrooms).    **Event Information Integration**   * The app must display details of campus events, including the time, location, and description. * Users should be able to navigate directly to event locations from the event details page. * Event notifications and reminders can be sent to users based on their preferences.    **Accessibility Features**   * The app must provide accessibility options, such as wheelchair-friendly routes and information on accessible entrances, elevators, and restrooms. * It should offer route recommendations based on accessibility needs.    **Geospatial Data Management**   * The app must integrate with a geospatial database (PostGIS with PostgreSQL) to store and manage campus map data, including coordinates for locations and paths. * The system should be able to perform spatial queries for calculating routes and identifying nearby resources.    **Backend and API Development**   * A backend server must handle data requests, user authentication, and map updates. * RESTful APIs should be developed to support communication between the frontend (React.js) and the backend (Django).    **Push Notifications**   * The app should send notifications for important updates, such as event reminders, location-specific alerts, or campus news. * Users should have the option to enable or disable specific types of notifications.    **Administrative Dashboard**   * An admin dashboard should be available for managing app content, including map updates, event scheduling, and user roles. * Administrators should be able to view analytics on app usage, popular routes, and frequently searched locations |
| User Interface |  **Login/Sign-Up Screen**   * A simple interface with options for logging in, signing up, and password recovery. * Social login options via OAuth 2.0 (Google, Facebook) for quick authentication. * Minimalistic design with a clean background, university logo, and form fields for input.    **Home Screen/Dashboard**   * A map-focused layout displaying the interactive campus map as the main element. * Quick access buttons for key features like "Search," "My Locations," "Events," and "Settings." * A bottom navigation bar or floating action button for frequently used functions, such as accessing saved locations or starting a navigation route.    **Interactive Map Screen**   * A full-screen map with zoom and pan controls, along with clickable markers for points of interest (buildings, facilities, events). * A search bar at the top for location lookup and filter options (e.g., dining, parking, academic buildings). * A "Locate Me" button to center the map on the user's current location. * Pop-up information cards that appear when a marker is selected, displaying details like the building name, hours of operation, or accessibility information.    **Navigation Screen**   * A map view showing the selected route with turn-by-turn navigation instructions. * Information on distance, estimated travel time, and alternate routes. * Options to switch between walking, biking, or accessible routes. * Real-time updates as the user progresses along the route.    **Search and Resource Finder Screen**   * A search input field with suggestions and auto-complete functionality. * Results displayed as a list below the search bar, with icons representing different categories (buildings, events, dining). * Ability to filter results by type (e.g., restroom, café, library).    **Events Screen**   * A list or calendar view displaying upcoming campus events, with clickable entries for more details. * Each event page should include the time, location, description, and a "Navigate Here" button. * Option to add events to a personal calendar or receive reminders.    **Settings Screen**   * Options for managing profile information, accessibility preferences, and notification settings. * Toggle switches for enabling or disabling specific map layers (e.g., parking zones, accessible routes). * Option to download the map for offline access.    **Admin Dashboard (Web Interface)**   * A web-based interface for administrators to manage map updates, user roles, events, and push notifications. * Data visualization tools for monitoring app usage, popular locations, and other analytics. |

|  |  |
| --- | --- |
|  |  |
| Content Management | * 1. **Campus maps**: Digital representations of building layouts, floor plans, outdoor areas and pathfinding.   2. **Point of interests(POIs):** Location such as lecture halls, library, board room, administrative offices, parking lots, and emergence services.   3. **Events:** Listings for campus events, workshops, seminars and other activities.   4. **Notifications and alerts:** important updates, emergence alerts and general campus announcements. |
|  |
| 6. Non-Functional Requirements |  |
| Performance | * **Response Time:** The app should load the main dashboard and map view within 2 seconds for the users with a stable internet connection. Navigation directions and Search results should be displayed within 1 second. |
| Scalability | * The system will support at least 10,000 concurrent users without a noticeable degradation in performance. |
| Security | * The app will use OAuth 2.0 for secure authentication, with support for multi-factor authentication(MFA) for increased security. |
| Compatibility | * The app will support the latest version of major mobile operating systems (Android and iOS), as well as web browsers (chrome, safari, Firefox). |
| Accessibility | * The app will comply with Web Content Accessibility Guidelines(WCAG) 2.1 Level AA to ensure that it is usable to people with disabilities. |
| Maintainability | * The app will be maintainable which encourages better performance of the system. |
| 7. Technical Requirements | * This section outlines the technical requirements for the campus navigation and resource finder app, covering architecture, technology stack, APIs, data management, and infrastructure. |
| Architecture | * **Modular design:** The app will follow a modular architecture to separate core functionalities (e.g., Navigation, Notifications, User Management). This will facilitate independent updates and scaling of each module. * Client server architecture: The app will follow a client-server architecture model where the frontend (mobile/web app) communicate with backend services via RESTful APIs. |
| Platform |  |
| Technology Stack | * Frontend (Mobile/Web): * Mobile Framework: Use React Native or Flutter for cross-platform development (Android and iOS support). * Web Framework: Use React.js for building the web version of the app. * Map Integration: Use Mapbox or Google Maps API for interactive maps and navigation features. * **Backend:** * Programming Language: Python (with Django or Flask for building the backend services). * API Framework: Django REST Framework or Flask-RESTful for creating RESTful APIs. * Authentication: OAuth 2.0 for user authentication and JWT (JSON Web Tokens) for session management. * Database: * Primary Database: PostgreSQL with GIS support using PostGIS for managing spatial data. * Caching: Use Redis for caching frequently accessed data to improve performance. * Hosting and Deployment: * Cloud Provider: AWS or Google Cloud Platform for hosting. * CI/CD: Implement Continuous Integration and Continuous Deployment pipelines using GitHub Actions, GitLab CI, or Jenkins. |
| API Requirements | **RESTful API Design:**   * Design APIs following REST principles, allowing the frontend to interact with the backend for data retrieval, updates, and deletions. * Ensure APIs are versioned to support backward compatibility.   **Third-Party API Integrations:**   * Map APIs: Integrate with Mapbox API or Google Maps API for geolocation, map rendering, and route calculations. * External Campus Systems: Integrate with existing campus systems (e.g., student information systems, event management systems) via SOAP or REST APIs.   **API Security:**   * Implement rate limiting to prevent abuse. * Secure all endpoints with OAuth 2.0 authentication. |
| Data Management | * Data Storage: Use PostgreSQL with PostGIS extension for spatial data storage. Store user profiles, event details, and POIs in relational tables. * Data Backup and Recovery: Perform automated daily backups of the database and support point-in-time recovery. * Data Retention Policy: Define a data retention policy for user data and logs, adhering to local data protection laws (e.g., GDPR). * Data Encryption: Encrypt sensitive data at rest and in transit (e.g., user passwords, personal information). * GIS Data Management: Use PostGIS to handle spatial queries, distance calculations, and route optimization. |
| 8. Timeline | The project timeline for the campus navigation and resource finder app is set to three months, with each phase carefully planned to ensure timely completion. |
| Milestones | 1. Milestone 1: Finalize PRD   Tasks: Conduct rapid stakeholder meetings, define user personas, document essential use cases, and complete the PRD with priority functional and non-functional requirements.   1. Milestone 2: Set Up Project Environment   Tasks: Establish project management tools, set up the development environment, and create the initial code repository.   1. Milestone 3: Complete Initial UI/UX Design   Tasks: Develop wireframes and high-fidelity prototypes for key features, including navigation and search functions.   1. Milestone 4: Design Review and Finalization   Tasks: Conduct quick user testing on prototypes, refine designs based on feedback, and finalize UI/UX for primary screens.   1. Milestone 5: Frontend and Backend Development Start   Tasks (Weeks 5-6): Begin frontend development, including UI elements and map integration, while simultaneously developing core backend services.   1. Milestone 6: Feature Integration   Tasks (Weeks 7-8): Integrate frontend with backend, connect features like navigation and resource finding, and set up authentication.   1. Milestone 7: Initial Testing   Tasks: Perform unit and integration testing, fix issues identified during testing, and gather feedback from a small beta group.   1. Milestone 8: Final Testing and Optimization   Tasks: Conduct performance testing, make final optimizations, and ensure readiness for deployment.   1. Milestone 9: Soft Launch   Tasks: Deploy the app to the production environment, perform a soft launch with limited users, gather feedback, and fix any critical post-launch issues.   1. Milestone 10: Full Launch   Tasks: Conduct a full-scale launch across campus, promote the app, and monitor performance and user feedback. |
| Phases | * Phase 1: Planning and Requirements Gathering (Weeks 1-2) * Phase 2: Design and Prototyping (Weeks 3-4) * Phase 3: Development (Weeks 5-8) * Phase 4: Testing and Quality Assurance (Weeks 9-10) * Phase 5: Deployment and Launch (Weeks 11-12) |
|  |
| 9. Budget | Given a tight budget limit of 500 ZMW, here’s an extremely minimal Budget section focusing on basic resources and free tools, prioritizing critical elements. The budget for the campus navigation and resource finder app is strictly limited to 500 ZMW. To keep within this limit, we will rely heavily on free tools, volunteer contributions, and minimal out-of-pocket expenses. |
| Cost Estimates and Resource Allocation | * + - Personnel Costs * Volunteer Work: All development, testing, and design work will be conducted by students, faculty, or volunteers. * Incentive for Contributors: * Basic stipend for top contributors: 200 ZMW   Total Personnel Costs: 200 ZMW   * + - Software and Tools       * + Free Design Tools: Use free versions of Figma, Adobe XD, or open-source design tools.         + Version Control and Collaboration: GitHub (Free Plan), Slack (Free Plan), Google Workspace.         + Map Services: Leverage free tier options like OpenStreetMap or limited usage with Google Maps API (Free Tier).         + Total Software and Tools Costs: 0 ZMW     - Infrastructure Costs       * + Free Hosting and Deployment: Use free-tier cloud services like AWS Free Tier or Google Cloud Free Credits for initial deployment.         + Database: PostgreSQL (Free and open-source) hosted on the cloud’s free tier.         + Total Infrastructure Costs: 0 ZMW     - Testing and Quality Assurance Costs       * + User Testing: Conduct testing within the campus community with volunteer participants.         + Basic Security Testing: Use free security scanning tools like OWASP ZAP for basic security checks.         + Total Testing Costs: 0 ZMW     - Miscellaneous Expenses       * + Marketing Materials: Social media promotion using free channels (e.g., campus bulletin boards, social groups).         + Miscellaneous Fund: 300 ZMW reserved for unforeseen small expenses, e.g., printing flyers or minimal online ads.         + Total Miscellaneous Costs: 300 ZMW         + Total Estimated Budget         + The total estimated budget for the campus navigation and resource finder app in Zambian Kwacha (ZMW) is:         + Personnel Costs: 200 ZMW         + Software and Tools Costs: 0 ZMW         + Infrastructure Costs: 0 ZMW         + Testing and Quality Assurance Costs: 0 ZMW         + Miscellaneous Costs: 300 ZMW         + Grand Total: 500 ZMW |

|  |  |
| --- | --- |
|  |  |
| 10. Risks and Assumptions |  |
| Risks | * Risks * Limited Budget: * Impact: Essential functionalities may be limited, potentially affecting user experience. * Mitigation: Prioritize core features, use free and open-source tools, and engage volunteer contributors to reduce costs. * Dependency on Free Tools and Services: * Impact: Potential limitations in features, storage, or usage that could affect the app’s performance and reliability. * Mitigation: Choose reliable free tools with good community support (e.g., OpenStreetMap for mapping), and monitor usage to stay within free-tier limits. * Data Accuracy: * Impact: Inaccurate or outdated GIS data could lead to navigation errors, impacting user trust. * Mitigation: Regularly update mapping data, rely on trusted open-source mapping services, and include a feedback feature for users to report inaccuracies. * Volunteer Commitment and Skill Gaps: * Impact: Dependence on volunteers may lead to delays or quality issues due to varying skill levels and availability. * Mitigation: Assign tasks based on skill level, provide mentorship where possible, and set realistic deadlines with flexibility. * Low User Adoption: * Impact: Limited user interest could reduce the app’s effectiveness as a campus resource. * Mitigation: Conduct pre-launch surveys to gather feedback, promote the app through campus channels, and encourage user engagement with incentives. * Lack of Technical Support: * Impact: Issues that arise may take longer to resolve without a dedicated support team. * Mitigation: Build a knowledge base for troubleshooting common issues and train volunteers to handle minor maintenance tasks post-launch. |
| Assumptions | Availability of Free Tools and Services:  Assumes that free-tier versions of necessary tools (e.g., GitHub, OpenStreetMap, and Google Cloud) will meet the project's needs without requiring upgrades.  Volunteer Workforce:  Assumes that student and faculty volunteers with the necessary technical skills will be available to contribute to development, design, and testing.  User Access to Internet and Smartphones:  Assumes that most users will have access to smartphones and reliable internet on campus, making the app accessible to the majority of the campus community.  Support from Campus IT:  Assumes that campus IT will support the project with access to existing campus data and resources, such as APIs for class schedules or resource availability.  User Interest and Engagement:  Assumes that there is interest among students and faculty for a campus navigation app, which will encourage adoption and feedback.  \*\*Long-term |
| 11. Approvals |  |
| Approval Signatures | **4. Approval Signature Example**  **Document**: Project Charter   | **Section** | **Details** | | --- | --- | | **Project Name** | Campus Navigation And Resource Finder | | **Project Manager** | Misheck Livingi | | **Project Sponsor** | Mbinji Nganga | | **Approval Date** | September 15, 2024 |   **Signatures**:   | **Name** | **Title** | **Signature** | **Date** | | --- | --- | --- | --- | | Misheck Livingi | Project Manager | [ ] | September 15, 2024 | | Mbinji Nganga | Project Sponsor | [ ] | September 15, 2024 | |