FORMATIVE ASSIGNMENT

ClassBeyond - *Learning that reaches every corner, beyond classrooms and borders.*

Description of the concept of their Mission;

My mission is to empower young minds in South Sudan and other refugee communities by providing inclusive, high-quality education to underprivileged children. Many children in conflict-affected and displaced settings lack consistent access to trained teachers, structured curricula, or learning materials. ClassBeyond is a digital learning platform designed to close this gap by offering accessible, curriculum-aligned content, interactive exercises, and mentorship opportunities tailored for children in refugee and low-resource settings.

This mission is rooted in my belief that education is the most powerful tool to break cycles of poverty, bridge opportunity gaps, and foster empathy for a peaceful future in South Sudan and across Africa.

Problem statement:

WHO: Children in refugee camps and marginalized rural areas across South Sudan and East Africa.
WHAT: Limited access to consistent quality education, learning resources, and teacher mentorship.
WHEN: An ongoing challenge, worsened during conflict, displacement, and crises such as COVID-19.
WHERE: Refugee settlements, IDP camps, and under-resourced rural schools.
WHY: Poverty, conflict, and displacement disrupt formal schooling and limit teacher availability. Many children lack access to digital learning opportunities that could bridge educational inequality.
HOW: By developing a digital platform (ClassBeyond) that offers offline/online learning resources, interactive lessons, teacher-student support, and progress tracking.

3. Prioritized Problems

- Lack of consistent access to qualified teachers
- Limited availability of learning resources and materials
- Disruption of education during displacement or emergencies

4. Proposed Solution

ClassBeyond is a mobile-first web application that:

- Provides curriculum-aligned digital lessons in Math, English, and Science.
- Works offline-first (downloadable lessons for areas with poor connectivity).
- Includes mentorship & peer-learning spaces where students can ask questions.
- Offers progress tracking dashboards for students, teachers, and parents.
- Integrates gamification (badges, quizzes) to increase student motivation.

Unlike generic e-learning platforms, ClassBeyond is designed for low-resource, refugee, and conflict-affected contexts, emphasizing offline access, multilingual support, and inclusive features.

5. Software Development Model

I will adopt the Agile Software Development Life Cycle (SDLC) model.

Why Agile?

- Education and refugee contexts require iterative feedback from teachers, students, and NGOs.
- Agile allows rapid prototyping, pilot testing, and continuous improvement.

Agile Steps I Will Follow:

1. Requirement Gathering: Interviews with refugee teachers, students, and NGOs.
2. Planning: Break the project into MVP features (basic lessons, login, offline mode).

3. Design: Wireframes for mobile-first UI, user stories, and system diagrams.

4. Development: Build MVP with HTML, CSS, JavaScript (frontend) + Firebase/Flask backend.
5. Testing: Pilot with small student groups in refugee schools.
6. Deployment: Use cloud hosting + offline mobile app builds.
7. Iteration: Refine features based on feedback (e.g., more languages, new subjects).

6. Hypothesis of the Solution

I hypothesize that;

If refugee and underprivileged children are provided with an accessible, offline-first digital learning platform, they will show measurable improvement in literacy, numeracy, and school retention rates, even in low-resource settings.

This will lead to a measurable increase in:

- Student engagement with learning resources
- Improved exam/test performance
- Higher continuity in education despite displacement
- Teacher and mentor participation in digital classrooms

7. References (APA Style)

- UNICEF. (2023). *Education in South Sudan*. https://www.unicef.org/southsudan/education

- UNHCR. (2022). *Stepping Up: Refugee Education in Crisis*. https://www.unhcr.org

- World Bank. (2021). *Learning Poverty in Developing Regions*. https://www.worldbank.org

- UNICEF. (2023). *Education in South Sudan*. UNICEF South Sudan. https://www.unicef.org/southsudan/education
  (Core background on the education crisis in South Sudan, with stats and challenges.)

- Gallagher, M. (2024). Hopeful futures for refugees in higher education: cultivation through connected learning and technology. *International Journal of Educational Technology in Higher Education*.

  https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00470-5
  (Explores refugee education through connected learning and tech, with Uganda case studies.)

- Rodríguez-Segura, D. (2022). *EdTech in developing countries: A review of the evidence*. World Bank.

  https://openknowledge.worldbank.org/entities/publication/2e40df91-909f-4cd2-888d-b9d3bf401f42
  (Strong evidence review of what works and what doesn't in EdTech for low-resource contexts.)

- Education for Life Project (Oxfam et al.). (2022). *Education for life: Well-being and resilience in South Sudan and Uganda*.
  https://www.ei-ie.org/file/108
  (Looks at resilience and well-being in refugee/host communities in South Sudan and Uganda.)

- Nicolai, S., et al. (2023). *Toward a holistic approach to EdTech effectiveness*. *Frontiers in Education*.
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10293898
  (Global open-access paper stressing context-based EdTech, relevant for your offline-first model.)

- Adediran, A. (2023). *Ed-Tech landscape and challenges in Sub-Saharan Africa*. Southern Voice.
  https://southernvoice.org/wp-content/uploads/2023/12/SV_Ops-N88-Pre.pdf
  (Policy brief focused on EdTech realities and barriers in Sub-Saharan Africa.)

Software Requirements Specification (SRS) Template

Title

ClassBeyond - *Learning that reaches every corner, beyond classrooms and borders.*

Prepared by Anthony Ariik Mathiang

Organisation: African Leadership University

Date created: 28th Sept 2025

Table of Contents.....

Revision History.....

1. Introduction.....

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements for ClassBeyond, version 1.0, a web-based learning platform designed to empower refugee and underprivileged children in South Sudan and other low-resource settings. The website provides curriculum-aligned lessons, interactive exercises, mentorship opportunities, and progress tracking accessible via desktop or mobile browsers, with offline functionality for low-connectivity areas.

This SRS covers the entire ClassBeyond website, including student learning modules, quizzes, mentorship scheduling, teacher content management, dashboards, and administrative tools.

1.2 Document Conventions

Requirement IDs: FR (Functional Requirement) and NFR (Non-functional Requirement) are uniquely numbered (e.g., FR01, NFR01).
Formatting: Headings in bold, key terms italicized.
Priorities: High-level requirements are assumed to cascade to sub-requirements unless specified.
Diagrams: UML diagrams follow standard Mermaid notation.

1.3 Intended Audience and Reading Suggestions

Developers: Implement the website features, offline caching, and responsive design.
Project Managers: Understand scope, milestones, and resources.
Testers: Design test plans for web interfaces, cross-browser compatibility, and offline functionality.
Teachers/Mentors/Students: Understand website capabilities.
Documentation Writers: Produce manuals, FAQs, and online tutorials.

1.4 Product Scope

ClassBeyond is a web-based digital learning platform designed to provide inclusive, high-quality education to learners in low-resource, refugee, and conflict-affected communities. It offers curriculum-aligned lessons in core subjects (Math, English, Science) and technology and professional development courses, combining academic knowledge with 21st-century skills.

The platform is offline-first, allowing learners to access downloadable lessons without internet connectivity. It features mentorship and peer-learning spaces, progress tracking dashboards for students, teachers, and parents, and gamification elements such as quizzes and badges to enhance engagement.

ClassBeyond aims to ensure continuous learning, foster skill development, and support educational inclusion, aligning with organizational goals to empower underserved communities through accessible and impactful digital education.

2. Overall Description

2.1 Product Perspective

ClassBeyond is a new, self-contained web platform. It can integrate with third-party APIs for authentication (Firebase/Auth0) and calendar scheduling.

2.2 Product Functions

- User registration and login (students, teachers, mentors, admins)
- Browse, download, and view lessons offline via browser cache or PWA
- Take quizzes and earn badges
- Request and schedule mentorship sessions
- Teacher content upload and progress monitoring
- Admin approval of content
- Notifications for updates and mentorship requests

2.3 User Classes and Characteristics

User Class	Characteristics	Frequency / Privileges
Student	Basic literacy, low technical expertise	Daily; access lessons, take quizzes, request mentorship
Teacher	Medium technical expertise, uploads content, monitors progress	Weekly; manage lessons and student dashboards
Mentor	Moderate technical expertise, guides students	Weekly; respond to mentorship requests

Admin	High technical expertise, manages system	As needed; approve content, manage users
NGO/Partner	Stakeholders monitoring impact	Monthly; view reports

2.4 Operating Environment

- Chrome, Edge, Firefox, Safari (desktop and mobile)
- Flask or Node.js AP
- PostgreSQL or Firebase
- Service Workers / IndexedDB for Progressive Web App (PWA) offline caching
- Offline access supported; periodic sync required

2.5 Design and Implementation Constraints

- Low bandwidth and intermittent internet
- Devices with limited RAM/storage
- PWA standards for offline-first access
- Multilingual support
- Compliance with child protection and data privacy regulations
- Use of standard web frameworks (React.js, HTML5, CSS3, JS)

2.6 User Documentation

- User manuals and guides for students, teachers, and mentors
- Online help and tutorials accessible from website
- Admin dashboard documentation

2.7 Assumptions and Dependencies

- Users have devices with modern web browsers
- Firebase/Auth0 available for authentication
- Teachers/mentors can provide content in supported formats
- Internet available periodically for syncing offline content

3. External Interface Requirements

3.1 User Interfaces

- Responsive web design for desktop and mobile
- Standardized buttons, menus, and navigation
- Error messages and tooltips
- Teacher/admin dashboards

3.2 Hardware Interfaces

- Devices: Desktop, laptop, tablets, mobile phones
- Optional microphone/webcam for mentorship
- Local browser storage for offline caching

3.3 Software Interfaces

- REST APIs for backend communication
- Firebase/Auth0 for authentication
- Google Calendar API for scheduling sessions
- PostgreSQL or Firebase database

3.4 Communications Interfaces

- Email and push notifications via browser
- Offline data sync via JSON over HTTPS
- SSL/TLS encryption for all communication

4. Requirement Specification

Stakeholder Requirements Specification

Functional Requirements

Req ID	Requirements	Description
FR 01	User Regis	Allow students, teachers, mentors, and admins to register and log in securely
FR 02	Browse & Download Lessons	Enable users to browse lessons and download them offline using PWA caching
FR 03	Quizzes & Badges	Allow students to take quizzes and earn badges for achievements
FR 04	Allow students to take quizzes and earn badges for achievements	Students can request and schedule mentorship with mentors
FR 05	Teacher Content Upload	Teacher can upload quizzes and lessons for review
FR 06	Admin Content Approval	Admin must review and approve uploaded content before publishing
FR 07	Progress Tracking	System tracks student progress and generate student report
FR 08	Notifications	Notify users about updates, new lessons, and mentorship requests
FR 09	Offline Sync	Sync offline data automatically once internet is restored

5. Other Nonfunctional Requirements

5.1 Example of Non-Functional Requirements

Requirement Type	Req ID	Description
Security	NFR 01	Authenticate all users using Firebase/Auth0
Performance	NFR 02	Support at least 100 concurrent web users
Usability	NFR 03	Provide an accessible interface suitable for low-literacy users
Auditability	NFR 04	All user actions logged for reporting

Requirement Type	Req ID	Description
Cross Browser Support	NFR 05	Functional on ● Google Chrome ● Microsoft Edge
Technology	NFR 06	Compatible with desktop and mobile browsers
Availability	NFR 07	Offline-first access enabled through PWA caching
Scalability	NFR 08	System should scale to 1000 additional users with minimal upgrades
Auditability	NFR 09	All user actions logged for reporting

6. Appendix

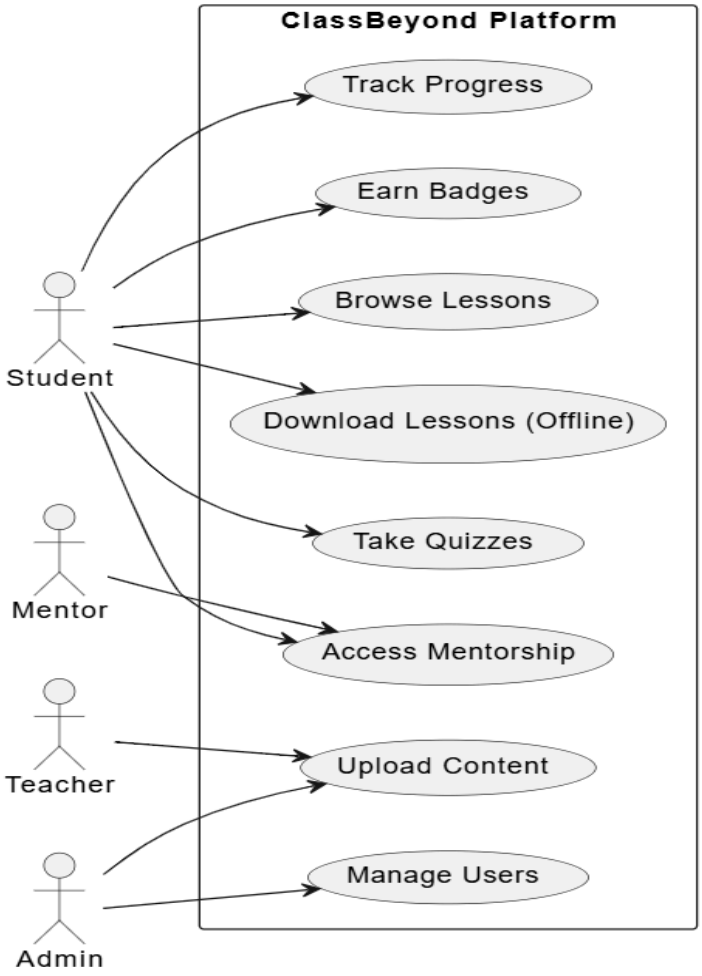
- MVP (Minimum Viable Product) – The initial functional version of the platform with core features such as offline lessons, quizzes, and mentorship.
- SDLC (Software Development Life Cycle) – The process framework guiding the development of ClassBeyond using Agile methodology.
- PWA (Progressive Web App) – A web application designed to work offline, installable on devices, and optimized for low-connectivity environments.
- Firebase – A backend-as-a-service platform providing authentication, database, hosting, and cloud services for ClassBeyond.
- Mentorship Session – A scheduled digital interaction between students and mentors for academic support.
- Gamification – The use of badges, points, and progress indicators to motivate student learning.
- Offline-first – A design approach ensuring ClassBeyond functions without internet connectivity and syncs data once reconnected.

Appendix B: Analysis Models

1. Use Case Diagram

Description

The Use Case diagram illustrates the functional requirements of the **ClassBeyond learning platform** by showing the interactions between different actors (users) and the system's use cases. It captures the main functionalities available to each user role and highlights how they engage with the system.



Explanation

Actors:

- Student: Primary user who accesses lessons, quizzes, and mentorship support.

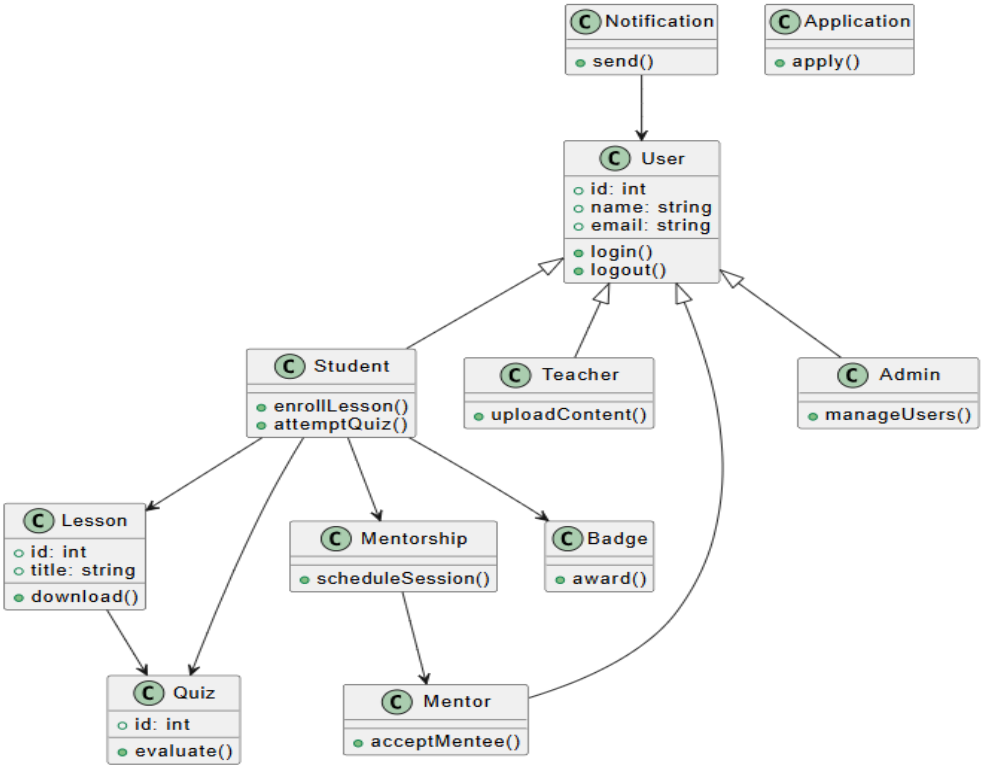
- Teacher: Educators who upload content, manage quizzes, and track student progress.
- Mentor: Volunteers/experts who provide mentorship sessions and feedback.
- Admin: System administrator responsible for approving content, managing users, and overseeing platform activities.

Key Relationships:

- Students interact extensively with the system by browsing/downloading lessons, taking quizzes, and requesting mentorship.
- Teachers mainly focus on uploading lessons, managing assessments, and tracking student performance.
- Mentors engage with the mentorship module by responding to student requests and scheduling sessions.
- Admins oversee all activities including user management and content approval.
- The "Download Lessons" use case incorporates both online streaming and offline-first access via caching.

2. Class Diagram

The Class diagram shows the static structure of the **ClassBeyond system** and describes the classes, their attributes, methods, and relationships. It represents the object-oriented design of the platform and indicates how different entities interact.



Explanation

Key Classes:

- User: Base class for common attributes (ID, name, email, login).
- Student, Teacher, Mentor, Admin: Inherit from User, each with role-specific attributes and methods.
- Lesson: Contains lesson ID, title, and downloadable/streaming capability.
- Quiz: Linked to lessons, includes questions, answers, and scoring functions.
- ProgressRecord: Stores a student's performance and tracks learning outcomes.
- MentorshipSession: Represents scheduled mentorship meetings between Students and Mentors.

Relationships:

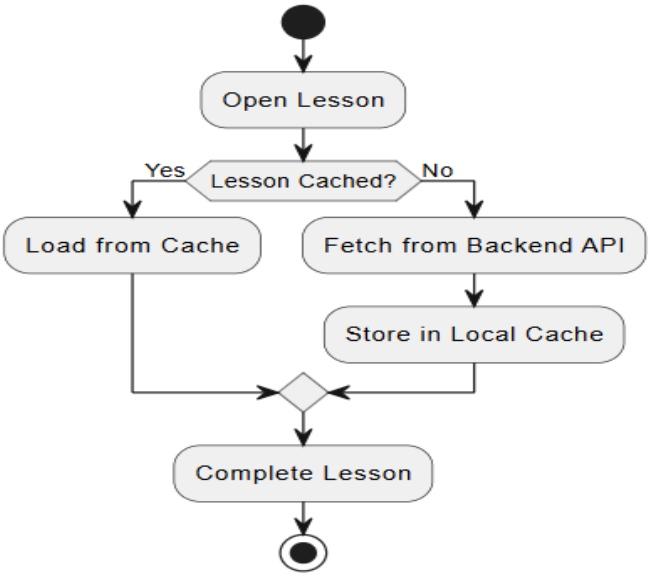
- Inheritance: Student, Teacher, Mentor, and Admin inherit from User.
- Association: Students are associated with Lessons, Quizzes, and ProgressRecords.

- Composition: A Quiz is composed of multiple Questions.
- Aggregation: MentorshipSession aggregates a Mentor and one or more Students.
- Dependency: Teacher depends on Lesson objects to upload or manage educational content.

3. Activity Diagram – Lesson Learning Process

Description

This Activity diagram shows the workflow of a student engaging with a lesson. It highlights the sequential steps a student follows, from logging in to completing a lesson, taking a quiz, and updating their progress.

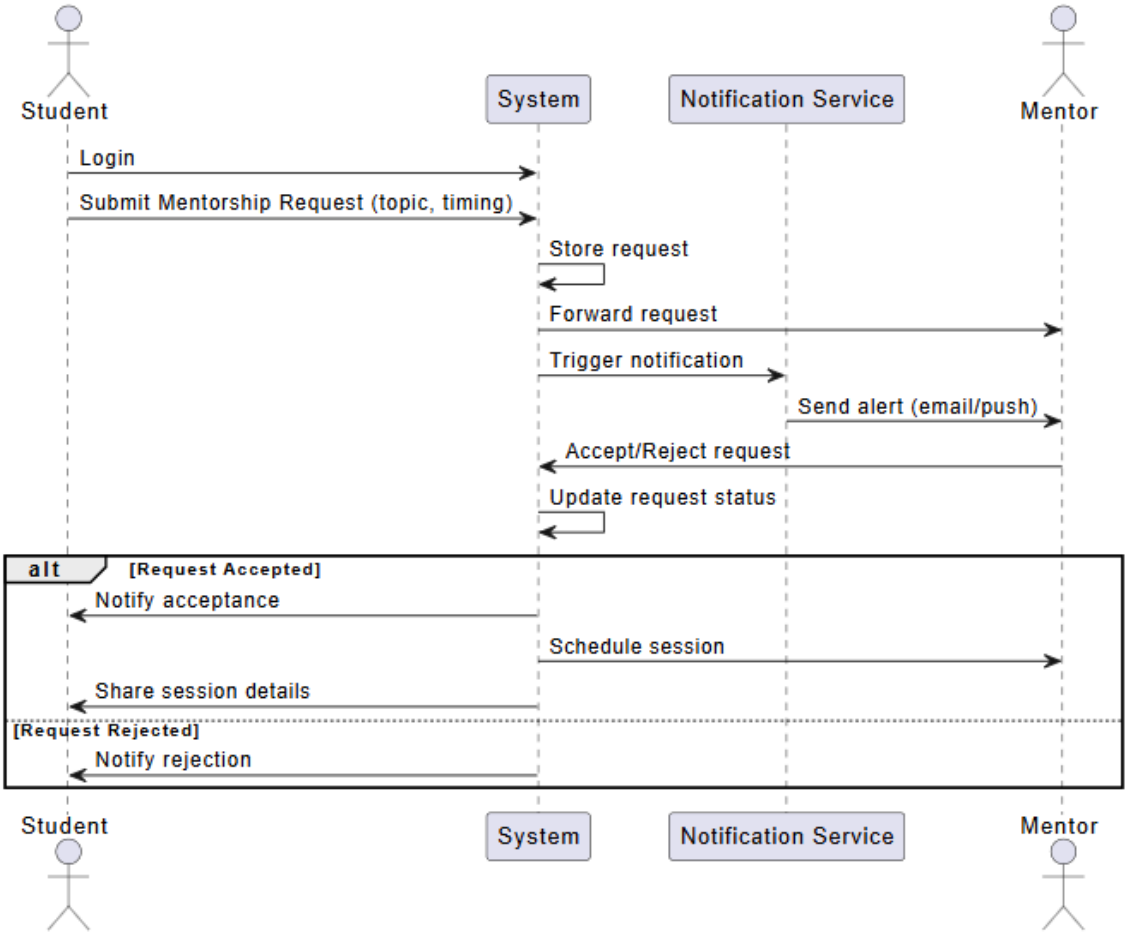


Workflow Includes:

- Login authentication
- Browsing or downloading lessons
- Decision node: check for internet connection → if offline, open cached lessons
- Taking the quiz linked to the lesson
- Awarding badges upon completion
- Updating the progress dashboard

4. Sequence Diagram – Mentorship Request Process

The Sequence diagram demonstrates the chronological interactions between student, mentor and the system during a mentorship request. It captures how a student submits a request, how the system notifies the mentor, and how scheduling is finalized.



Explanation

Actors and Objects:

- **Student:** Initiates the mentorship request.
- **System:** Manages request submission, notifications, and scheduling.
- **Mentor:** Reviews and accepts/rejects requests.
- **Notification Service:** Delivers alerts via email or push notifications.

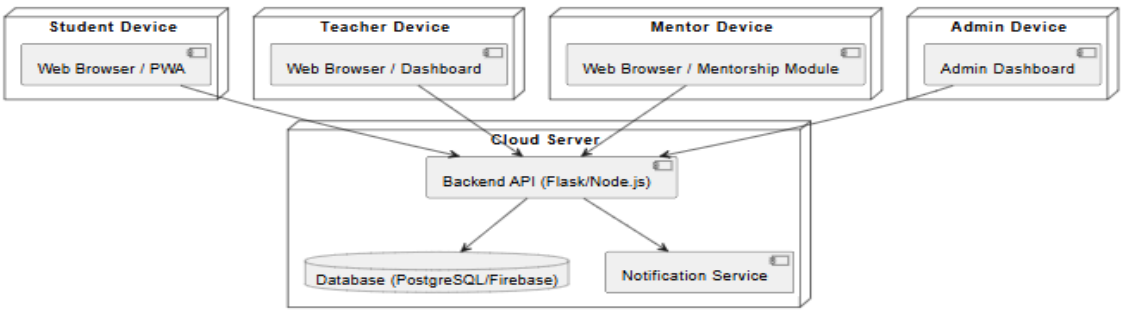
Message Flow:

1. Student logs into the system.
2. Student submits mentorship request with topic and timing preferences.
3. System stores the request and forwards it to the Mentor.
4. Notification Service sends an alert to the Mentor.
5. Mentor accepts/rejects the request.
6. System updates the request status and notifies the Student.
7. If accepted, System facilitates session scheduling between Student and Mentor.

Key Interactions:

- System acts as a mediator between Student and Mentor.
- Notifications ensure asynchronous communication.
- Student receives real-time updates about mentorship status.

5. Deployment Diagram



Summary

These four UML diagrams provide a comprehensive architectural and functional view of the **ClassBeyond learning platform**:

- Use Case Diagram: Defines user interactions and system functionalities.
- Class Diagram: Outlines the platform's object structure and relationships.
- Activity Diagram: Shows the step-by-step learning process with offline/online decisions.
- Sequence Diagram: Demonstrates temporal flow of mentorship request handling.