# **Group username: dr.kimmoos**

# **KU volunteer:**

## **1. Background**

Kasetsart University students must complete a required number of volunteer hours across different categories in order to graduate. Currently, volunteer activities are scattered across multiple channels such as faculty notice boards, club social media pages, and printed flyers. This fragmented process makes it difficult for students to discover and track available opportunities.

For organizers and university staff, managing activities is also inefficient. Each new event often requires creating separate registration forms, manually tracking attendance, and verifying student participation. This results in duplicated effort and limited transparency.

To address these challenges, the KU Volunteer system will provide a centralized, web-based platform where students can easily find activities, submit applications, and track their progress, while organizers and staff can manage events and verify hours more effectively.

## **2. Stakeholders**

**Student**:

* Search/filter activities → via keyword search, category tags, and date filters.
* Apply for activities → standardized digital application form stored in database.
* Upload proof → upload documents/images (stored in S3 / Django file storage).
* Track hours → dashboard view pulling from verified attendance records.

**Organizer**:

* Create/edit activities → activity creation form (title, description, category, dates, max participants).
* Add custom questions → dynamic form builder with text/MCQ fields.
* Manage capacity → system automatically disables registration once capacity is reached.
* Approve/reject applications → dashboard with pending applicants, one-click approval.
* Mark attendance → QR code check-in or manual attendance list.

**Staff/Admin**:

* Verify hours → access to all attendance logs + override button.
* Audit logs → backend logs of all approvals, rejections, and overrides.
* Generate reports → export to CSV/PDF by student, activity, or category.

## **3. Objective**

The main goal of the **KU Volunteer**  is to fix the current problems by creating one simple online system for all volunteer activities.

* **Provide one platform for all volunteer opportunities:** This web-based application will solve the problem of information being spread out in many different spots, making sure all students can see and access every available activity.
* **Help students track required hours by category:** To give students easy-to-use tools to keep an eye on their required hours. This means showing them a clear and up-to-date view of how many hours they have finished and how many they still need for each type of volunteering, helping them make good choices about which events to join.
* **Improve transparency and management for staff** : To greatly reduce the extra work for event organizers and university staff by handling jobs like sign-ups, attendance lists, and checking hours automatically.

## **4. Detail of Implementation**

## **4.1 Proposed Features**

**Students** will be able to search for available activities, submit applications, and track the status of their participation through the system.

**Organizers** will have the ability to post and manage activity details, review student applications, and approvesubmissions, ensuring smooth coordination and efficient management of activities.

### **4.2 Technology Stack**

**Frontend:**

Language: Typescript, Javascript

UI : Nextjs, React, Tailwind, Figma

**Backend:** Django + DjangoREST Framework

**Database:** PostgreSQL

**Deployment:** **Containerization:** Docker & Docker Compose (dev/staging/prod)

**Reverse proxy:** NGINX or TraefikRender

**TLS**: Caddy or NGINX + Let’s Encrypt

**CI/CD**: GitHub Actions (lint, test, build, push, deploy)

**Monitoring**: Sentry (FE/BE), Prometheus + Grafana, Healthcheck endpoints

## **5. Member lists**

1. 6610545235 - Yatichapat Dechaeamsakul
2. 6610545278 - Natawipa Poonyakariyakorn
3. 6610545952 - Swischya Sunthonphusit
4. 6610545545 - Supidcha Nilsarika

## **6. Design & Technical**

* Architecture/Design updates:
  + Choose **Next.js + TailwindCSS** for frontend.
  + Selected **Django + Django REST Framework** with PostgreSQL for backend.
* API / Interface changes:
  + None yet (setup sprint).
* Data model changes:
* Other technical notes
  + Configured Docker & Docker Compose for local dev environment.:

## **7. Planning & Tracking**

## Sprint Number & Dates: Sprint 1– 25/08 to 07/09

* Sprint Goal: Set up initial project structure and development environment.
* Selected Stories / Epics (Jira IDs):
  + KV-40 As a developer, I want to set up the project structure so we can start development smoothly.
  + KV-41 As a system admin, I want a reliable deployment pipeline so I can ensure the app runs consistently.
  + KV-71 As a student/organizer/staff, I want clear and simple UI screens so I can easily use the volunteer system.
* Story Points Committed: [XX]
* Assignees (summary)
  + Supidcha Nilsarika -> Frontend & Backend initialization, Docker setup.
  + Natawipa Poonyakariyakorn -> GitHub repo & Kanban board.
  + Swischya Sunthonphusit -> UX/UI Figma design
  + Yatichapat Dechaeamsakul -> UX/UI Figma design
* Dependencies / Risks:
  + CI/CD pipeline setup depends on access to staging environment.

## **8. Sprint Review & Retrospective**

**Sprint Review:**

* Planned vs Completed:
  + Completed:
    - Initialize Django
    - Initialize Next.js
    - Configure docker
    - Set up CI/CD
    - Create UI mockup
    - Create Ux wireframe
    - Prepare design handoff
  + In Progress:
  + Not Started:
* Demo / Feedback:
  + Showed a basic Django backend with working endpoints.
  + Displayed the Next.js frontend with placeholder pages.
  + Confirmed CI/CD pipeline builds and deploys automatically.
  + Shared UI mockups and UX wireframes for feedback.
  + Feedback was positive on clear design and smooth setup.
  + Suggestions:
  + Make design handoff docs more consistent.
  + Add a sample end-to-end flow (UI → API → database) in the next sprint.

**Sprint Retrospective:**

* ✅ What Went Well:
  + Set up the main tools (Django, Next.js, Docker, CI/CD) successfully.
  + Frontend, backend, and design teams worked well together.
  + Got early feedback from mockups and wireframes.
  + No major problems came up during setup.
* ⚠️ What Can Be Improved:
  + Make the handoff between design and development smoother.
  + Write clearer and more detailed task requirements.
  + Share progress updates more clearly with stakeholders (for example, using a sprint dashboard).
* 💡 Action Items for Next Sprint:
  + Add Google login with email option as backup.
  + Set up user roles (Student, Organizer, Staff).
  + Lock backend APIs so only the right roles can use them.
  + Make login/signup forms on the frontend with checks for valid input.
  + Build a simple starting version of the staff admin dashboard.

## **9. Supporting Documents**

* Test Cases / Test Reports: None yet (initial setup sprint).
* Bug Reports (Jira IDs): None logged.
* QA Notes: Basic dev environment tested successfully on 2 machines.

## **10. References & Links**

* Jira Sprint Board: <https://ku-team-f030w3d7.atlassian.net/jira/software/projects/KV/boards/35?jql=assignee+IN+%28712020%3Acd589b99-7de7-4cbb-bc6c-cd7861216596%2C+712020%3Ae2d51e54-1c72-4356-8b39-40665465483f%2C+empty%29&atlOrigin=eyJpIjoiNzIyM2I1M2YyYmUyNDRkYjlkZWJhZjJhMTIwYzEwZjMiLCJwIjoiaiJ9>
* GitHub Repo: <https://github.com/natawipa/ku-volunteer>
* CI/CD Logs or Notes: Not yet
* Youtube review link
  + Sprint 1: https://youtu.be/TEeooISUzhA
* Other Resources:
  + Figma : <https://www.figma.com/design/xlDIr7uXlz8cMf5fttJkrg/KU-Volunteer?node-id=0-1&p=f&t=SWZXh5FlLdzx9iMA-0>