Department of Computer Engineering, Bilkent University
CS319 Object Oriented Software Engineering
Group 11
Final Report (Including User Manual, build instructions and workload allocation) - D5
Section 1 - Eray Tüzün
11/05/2025

Furkan Mert Aksakal 22003191
Furkan Komaç 22102165
Alper Biçer 22003097
Erkan Can Arslan 22103948
Deniz Yazıcı 21902557

Contents

Contents **2**
User Manual **3**
1. Introduction 4
2. Getting Started 4
    2.1 Logging In 4
    2.2 Navigation and Layout 4
3. User Roles and Access 5
4. Dashboard Overview 5
    4.1 TA Dashboard 5
    4.2 Instructor Dashboard 5
    4.3 Other Dashboards 5
5. TA Request Management 6
    5.1 Viewing Requests 6
    5.2 Creating TA Requests 6
    5.3 Editing Requests 6
    5.4 Request Details Page 6
6. Duties Management 7
    6.1 Viewing Duties 7
    6.2 Creating Duties 7
    6.3 Editing Duties 7
    6.4 Assigning TAs 7
    6.5 Logging Duties (TA) 7
    6.6 Approving Logs (Instructor) 7
7. Proctoring Management 8
    7.1 Viewing Assignments 8
    7.2 Auto vs Manual Assignment 8
8. Leave Management 8
    8.1 TA Leave Request 8
    8.2 Viewing Status 8
    8.3 Managing Requests 8
9. User Management 9
    9.1 Viewing Users 9
    9.2 Creating Users 9
    9.3 Editing Users 9
10. System Administration 9
    10.1 Admin Dashboard 9
    10.2 Data Import 9

10.3 System Logs..... 10
2. Build Instructions..... 11
Prerequisites..... 11
Step 1: Clone the Repository..... 11
Step 2: Configure Environment Variables..... 11
Step 3: Build and Start the Docker Containers..... 12
Step 4: Run localhost..... 12
Step 5: Create a Superuser (Optional)..... 12
Step 6: Import Initial Data (Optional)..... 12
Step 7: Access the Application..... 13
Managing the Application..... 13
System Requirements..... 13
Troubleshooting..... 14
Common Issues..... 14
3. Work Allocation..... 15

User Manual

1. Introduction

The TA Management System is an all-in-one platform developed for managing Teaching Assistants within Bilkent University. The system provides role-based functionality tailored for TAs, Instructors, Secretaries, Department Chairs, Deans, and System Admins.

Key Goals:

- •Streamline TA assignments and proctor scheduling
- •Enable TAs to submit logs and manage duties
- •Provide role-based dashboards and workflows

2. Getting Started

2.1 Logging In

Go to TA Management System login page.
Enter your username (usually your email) and password.
Click Login. You will be redirected to your personalized dashboard.

2.2 Navigation and Layout

- Top Navigation Bar: Access sections like Dashboard, Duties, Proctoring, and Requests.
- Responsive UI: Works on desktops and tablets; some features may have limited support on mobile.

3. User Roles and Access

Role	Main Permissions
TA	Log duties, request leaves, view assignments
Instructor	Manage TA duties, approve logs, assign TAs
Secretary	Manage accounts, view reports, process TA requests
Dept. Chair	Assign roles, view department-wide stats
Dean	Access faculty-level reports
Admin	Manage users, configure system-wide settings

4. Dashboard Overview

4.1 TA Dashboard

- Duty Logs: Submit logs for completed work
- Leave Requests: Apply for time off
- Swap Requests: Request changes in proctoring schedules

4.2 Instructor Dashboard

- TA Requests: Request new TAs for upcoming semesters
- Course Management: Create and manage all types of duties
- Log Approvals: Review and approve TA submissions

4.3 Other Dashboards

Each dashboard presents cards or tabs based on role:

- Secretary: TA requests, user approvals
- Department Chair: Stats, user role assignments
- Dean: Faculty-wide reports
- Admin: System logs, imports, global configs

5. TA Request Management

5.1 Viewing Requests

- Go to TA Requests > My Requests
- Click any request to view full detail, status, and feedback

5.2 Creating TA Requests

1. Navigate to TA Requests > New Request
2. Fill the form:
    - Select Course
    - TA Load: Minimum and maximum hours per week
    - Graders: How many grading assistants are needed
    - TA Preferences:
    - Must-Have TAs: With justification
    - Preferred TAs / Graders
    - To Avoid TAs / Graders
3. Submit the request

Example: If you had a great TA last semester, add them under “Preferred TAs.”

5.3 Editing Requests

- Go to My Requests
- Click “Edit” next to the relevant request
- Make changes and resubmit

5.4 Request Details Page

- Shows semester, instructor, workload, TA/grader requirements, and justification
- Linked to course offering object

6. Duties Management

6.1 Viewing Duties

- Go to Manage Duties > List Duties
- Use filters for course, duty type, or status (Pending, Approved, etc.)

6.2 Creating Duties

1. Go to Create Duty
2. Enter:
    - Course Offering
    - Date, Start Time, End Time
    - Duration (in hours)
    - Description
3. Type-specific fields:
    - Lab: Lab session number
    - Grading: Midterm/Final/Homework
    - Recitation: Topic
    - Office Hour: Room
    - Proctoring: Exam reference + classroom

6.3 Editing Duties

- Click a duty from list
- Make edits and save

6.4 Assigning TAs

- On Duty Detail Page, click Assign TA
- Choose from dropdown of available TAs
- Click Confirm Assignment

6.5 Logging Duties (TA)

- Go to Log Duty
- Choose duty and enter time spent (e.g. 2h 30m)
- Submit

6.6 Approving Logs (Instructor)

- Go to TA Management > Approve Logs
- Click Approve or Reject

•Add feedback when rejecting

7. Proctoring Management

7.1 Viewing Assignments

•Go to Manage Proctoring
•Statuses: Unassigned, Partial, Complete

7.2 Auto vs Manual Assignment

•Auto Assign:
•Click Auto Assign from exam list
•Select exam, set number of TAs, and generate automatic assignment
•Manual Assign:
•Select exam, pick TAs from dropdowns
•Submit

8. Leave Management

8.1 TA Leave Request

1. Go to Request Leave
2. Select start and end dates
3. Provide reason and notes
4. Submit

8.2 Viewing Status

•Go to Requested Leaves
•Status: Pending / Approved / Rejected

8.3 Managing Requests

•Instructors/Admins: Go to Manage Leaves
•Click Approve/Reject, provide feedback

9. User Management

9.1 Viewing Users

- •Go to Manage Users
- •Filter by role, status, or department

9.2 Creating Users

1. Click New User
2. Enter:
   - •Username, email, name
   - •Role and department
   - •Password (or auto-generated)
3. Submit

9.3 Editing Users

- •From list, click Edit
- •Change roles, contact info, department
- •Save

10. System Administration

10.1 Admin Dashboard

- •Access full system settings
- •View data stats and configuration options

10.2 Data Import

- •Go to Import Data
- •Upload .csv for:
- •Courses
- •Students
- •Instructors
- •Validate preview and confirm

10.3 System Logs

- Navigate to System Logs
- View activity logs
- Filter by user/date/module
- Export logs as .csv or .xlsx

2. Build Instructions

This guide will walk you through the process of installing and running the TA Management System locally on your computer. The system is built using Django, MySQL, and Docker.

Prerequisites

Before beginning, ensure you have the following installed:

1. Docker - For containerization
2. Git - For cloning the repository

Step 1: Clone the Repository

```
git clone https://github.com/yourusername/CS319---Group-11-Term-Project.git
cd CS319---Group-11-Term-Project/code/ta_mng_sys
```

Step 2: Configure Environment Variables

Create a `.env` file in the `ta_mng_sys` directory with the following content:

```
# Database configuration
POSTGRES_DB=tamng
POSTGRES_USER=tamnguser
POSTGRES_PASSWORD=strongpassword
POSTGRES_HOST=db
POSTGRES_PORT=5432

# Django configuration
DEBUG=True
SECRET_KEY=your-secret-key-here
ALLOWED_HOSTS=localhost,127.0.0.1
```

Replace your-secret-key-here with a random secure string.

Step 3: Build and Start the Docker Containers

From the ta_mng_sys directory, run:

```
make build
```

This command will build the Docker containers according to the docker-compose.yml file. It might take a few minutes the first time as it downloads and installs all dependencies.

Step 4: Run localhost

Connect to your localhost on port 8000.

Step 5: Create a Superuser (Optional)

Create an admin user to access the Django admin interface:

```
make superuser
```

Follow the prompts to create a username, email, and password.

Step 6: Import Initial Data (Optional)

You can import sample data for courses, instructors, students, and TAs using the provided functionality. After running the system, navigate to the data import page (accessible via the admin menu).

Sample Excel files are available for download on the import pages:

- Courses
- Faculty (Instructors)
- Students
- Course Offerings
- Enrollments

Step 7: Access the Application

The application should now be running. Open a web browser and navigate to:

```
http://localhost:8000
```

To access the admin interface, go to:

```
http://localhost:8000/admin
```

Log in with the superuser credentials you created earlier.

Managing the Application

Here are some useful commands for managing the application:

- Start containers: `make up`
- Stop containers: `make down`
- Restart containers: `make restart`
- View logs: `make logs`
- Run Django shell: `make shell`
- Run tests: `make test`
- Create new migrations: `make makemigrations`
- List running containers: `make list`

System Requirements

- Docker with at least 2GB of allocated memory
- At least 1GB of free disk space
- A modern web browser (Chrome, Firefox, Safari, or Edge)

Troubleshooting

Common Issues

1. Database connection errors
    - Make sure the database container is running: `docker-compose ps`
    - Check database logs: `docker-compose logs db`

2. Permission issues
    - Some operations might require admin privileges. Make sure you're using an admin account.

3. Static files not loading
    - Run make collectstatic again
    - Clear your browser cache

4. Port conflicts
    - If port 8000 is already in use, you can modify the port mapping in docker-compose.yml

3. Work Allocation

- Alper Biçer
    - Implemented the module that evenly assigns students into the exam classrooms alphabetically.
    - Implemented the manual TA assignment to the courses. Single assignment can be made.
    - Implemented the TA assignment to the courses with an excel file.
    - Implemented the feature that instructors can view their TAs' total workload and current workload.
- Deniz Yazıcı
    - Implemented Excell importing module. Enabled user to populate system with needed data.
    - Implemented automatic proctor assignment module for exams.
    - Implemented manual proctor assignment module for exams.

- Furkan Komaç
    - Making full implementation of Leave Request model. (TA, Instructor interfaces etc.)
    - Making full implementation of Swap Request model. (TA, Instructor interfaces etc.)
    - Making full implementation of Log Duty model. (TA, Instructor interfaces etc.)
    - With Erkan Can Arslan, writed the code of every model (class) in the program.

- Erkan Can Arslan
    - Created the project framework base. Containerized project with Django and MySQL and prepared makefile for ease of development process.
    - Created the base user interface components of the application that has been used program-wide. Role-aware navbar structure is created.
    - With Furkan Komaç, writed the code of every model (class) in the program.
    - Made full implementation of TA Request module for CS department.
    - Made full implementation of Duty Management module.

- Furkan Mert Aksakal
    - Making full implementation of Manage User model. (CRUD of users by admin)
    - Making full implementation of System Logs (Seeing all the information and updates about the system and the users made)