Students:

Antonino Foti - 1799048
Michael Wijesinghe Arachchige Don - 1845004
Valerio Liberati - 1849580
Leonardo Sereni - 1846461
Mario Morra - 2156770

Project proposal:

# Fix This Stuff

## Introduction

The project is a customer service type web application. It can be used by **users** who can upload coding issues via ticketing system and solve the proposed coding problems, along with the possibility to create tickets themself. **Tickets** created by users have a list of properties like a description/main topic and are assigned to developers by a third figure called **moderator**, who can view all the pending tickets and the available developers before deciding how to assign tickets by looking at the ticket details and developer skills and ranking. Moderators cannot register themselves into the application since their profile can be only created by the **administrator**(s) of the system.

## User Case Scenario

- **Case 0:** (prerequisite) the administrator opens the application and creates a new moderator specifying name, credentials and so on.

- **Case 1**: an user opens the application and registers to the portal. The system asks for email, password, name, surname, role and so on and creates the user profile.

- **Case 2:** A user can create a ticket describing the issue and setting topics, title, priority and deadline. The customer can view his currently opened and past (closed) tickets and they can read comments or write other comments for any ticket. Before creating a ticket, a customer can navigate the other users (developers) page to see if there is already any similar ticket already solved.

- **Case 3:** A user (developer) can create a ticket describing the issue and setting topics, title, priority and deadline. The user (developer) can view his currently opened and past (closed) tickets and they can read comments or write other comments for any ticket.

- **Case 4:** moderator views the list of opened tickets and the list of available developers. Moderator can open a ticket to see details and assign it to a specific developer that matches the customer needs. Moderator can also decide to close a ticket if he sees that there is another ticket, already solved, about the same problem.

- **Case 5:** a registered user (developer) reads reports sent from users and assigned from moderator, then they can send back content with solved issues with comments if needed.

- **Case 6:** a registered user (developer) reads a report assigned from the moderator, then he can forward the issue to another user (developer) through the ticketing system and send back the solved issue to the moderator with comments if needed.

- **Case 7:** moderator receives a solved issue sent from a user (developer )and he can flag the ticket as “solved” and they send back contents to the user with comments if needed.

- **Case 8:** a moderator can log in the system and open a section, hidden to users, showing statistics about developers and ticket solving stuff.

# Operational Diagram

*DISCLAIMER: manage == view, edit, remove*

- Admin
  - Login/logout operation
  - Manage users
  - Manage developers
  - Manage customer service users
  - Manage tickets
  - View and edit/remove content

- User
  - Register/login/logout operation
  - Ticket creation
  - Access opened ticket history
  - View assigned tickets
  - Upload content
  - Deliver fixed part to user (customer) service
  - Open tickets for other users
  - Access history of fixed tickets

- Moderator
  - Login/logout operation

    - Change issue data sent by standard user from natural language to technical language for developers
    - Manage tickets and assign a developer to solve an issue
    - Manage flags based on priority
    - Assign priority
    - View users/tickets
    - View users (developer) statistics and ranking system

# Class Diagram

- Ticket properties
    - Number/id
    - Priority (low/medium/high)
    - Topics (e.g. python, database)
    - Title
    - Request field
    - Answer field
    - Request author
    - Assigned developer
    - Flag (open/closed) (solved/not solved)
    - Comments
    - Opening date
    - Deadline

- Administrator properties
    - ID
    - Name
    - Surname
    - Credentials

- Moderator properties
    - ID
    - Name
    - Surname
    - Credentials

- Developer properties
    - Name
    - Surname
    - ID
    - Rank/number of solved issues
    - Credentials
    - Date of registration
    - Skills (e.g. python)
    - List of opened tickets  (open/closed) (solved/not solved)

# Technologies Specifications

- Backend
    - PostgreSQL DB
    - Docker/Kubernetes containers to deploy
    - NodeJS
    - Apache

- Frontend
    - Bootstrap
    - React

- Middleware
    - RabbitMQ
    - Spring Boot

- Management tools
    - GitHub
    - Google Docs
    - Balsamiq