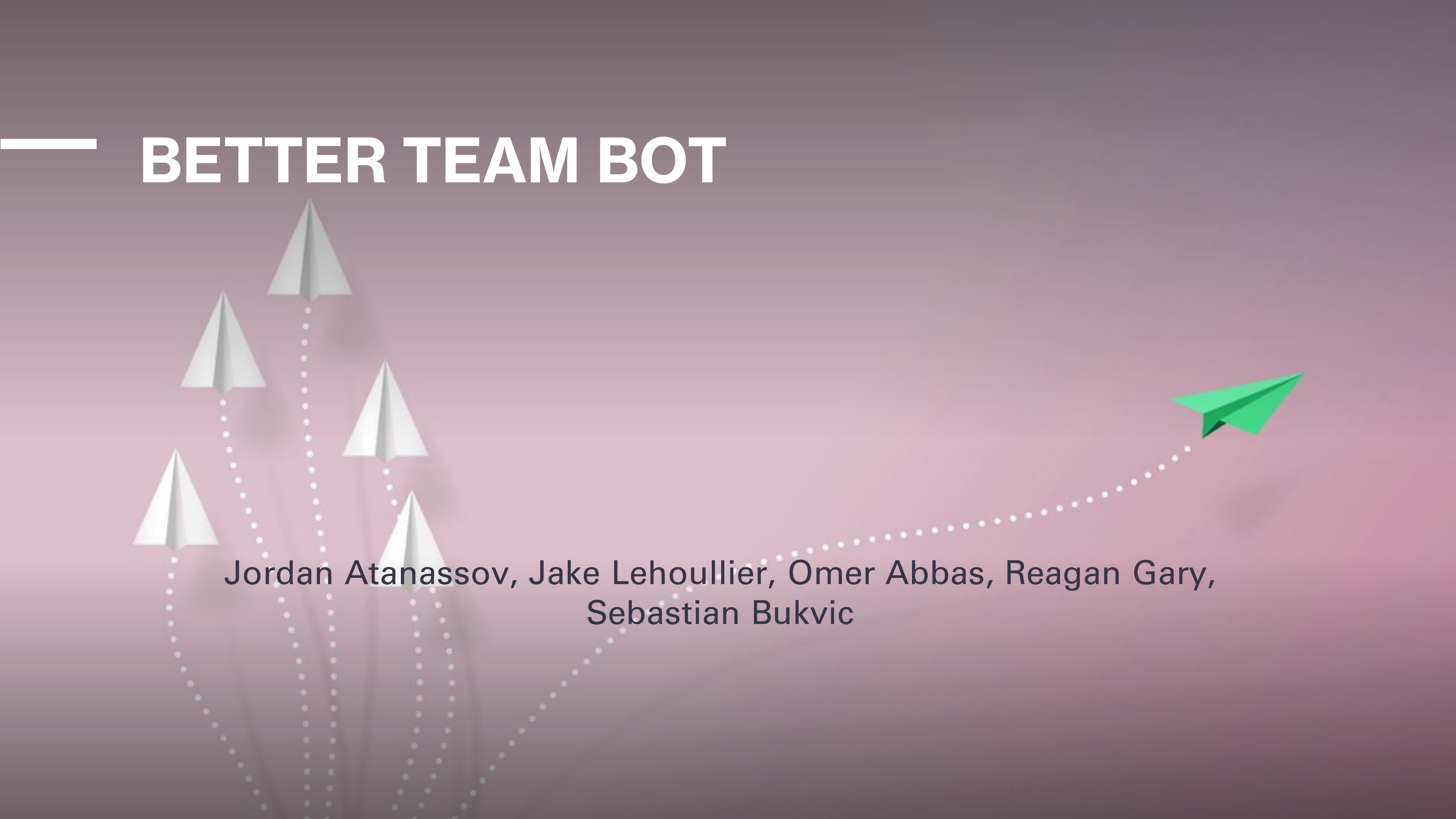
BETTER TEAM BOT



Jordan Atanassov, Jake Lehoullier, Omer Abbas, Reagan Gary, Sebastian Bukvic

Problem Statement

- In IRAD's (internal research and development projects), new-commers onto a project will not know exactly how to contribute to the project, not knowing what their role, their impact, and necessarily what doing their tickets adds to the project in the grand scheme.

Proposed Solution

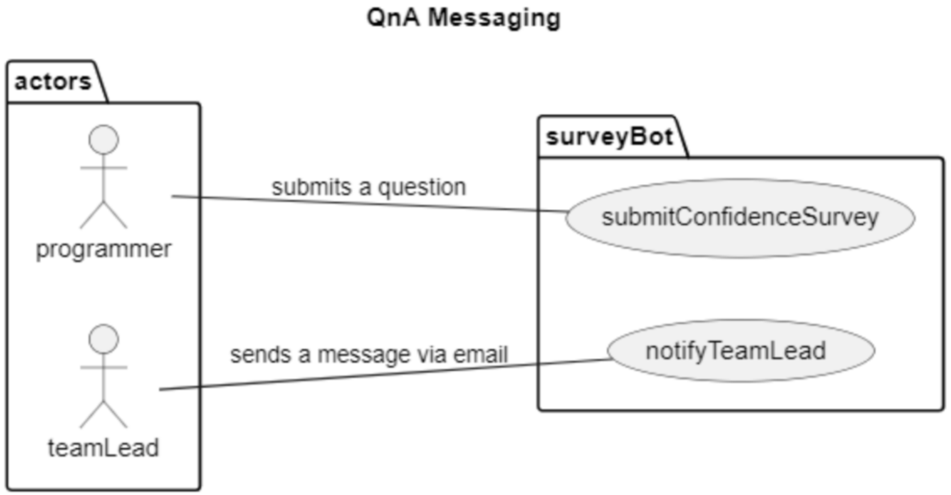
- The solution to the problem is our project, the Better Team Bot
- Better Team Bot is a transcript summary bot used in team meeting to make the meeting go smoother
- Records the meeting and provides a transcript afterwards
- Sends out a survey to all meeting participants after the survey to collect feedback, thoughts, and progress on projects
- Allows newcomers on a project to get caught up with what's going on and provide feedback if they are lost or confused

Rational [1]

- University of South Carolina states that the biggest drivers of software productivity, is that the staff is in small, well-organized teams, large teams should be broken down, and there should be short development schedules.
- It is hard to accurately know what will make it out of the research phase and be implemented into a projects long term lifespan
- They also state that it is incredibly hard to be productive when designing something from the ground up or starting from nothing, which IRADS aim to do in order to create something new.

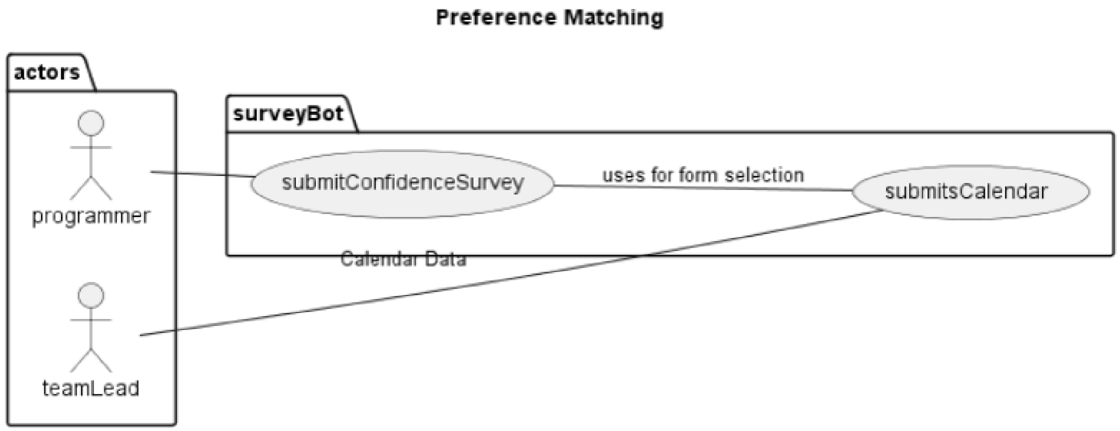
Use Case 1

- Pre-Condition
  - User attends a meeting
  - User misunderstands something in the meeting
- User Submits a survey
  - Sub-flow
    - User logs into the bot and submits questions
    - When a message is submitted and confidence is rated, the bot will then decide whether or not to notify the project manager
    - Project Manager then gets to respond
  - Alternate
    - Model deems the question not pertinent
    - Multiple people submit similar questions, and a group session is created

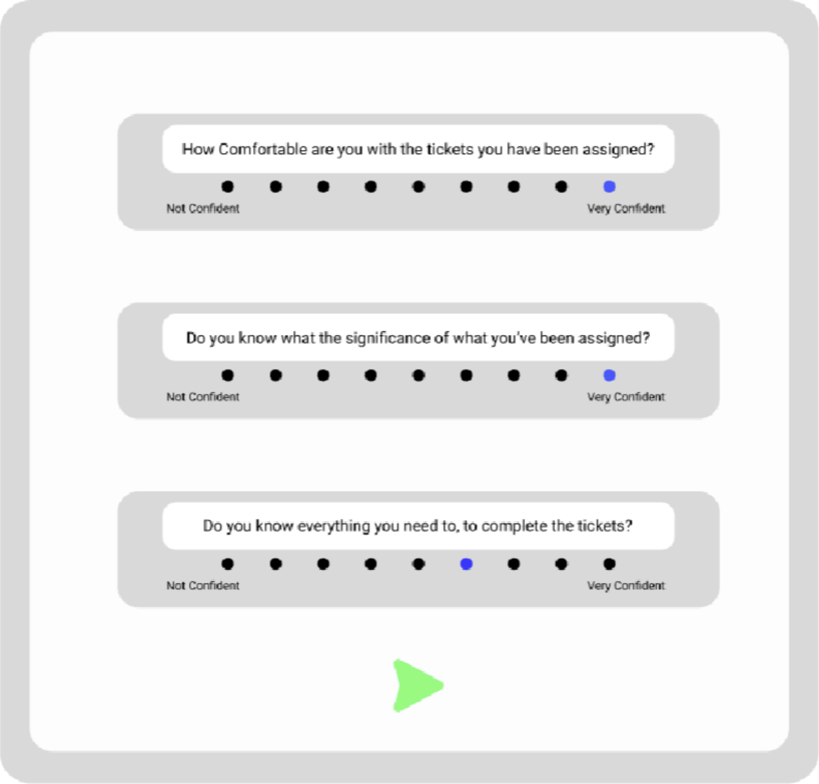


Use Case 2

- Pre-Condition
  - User first signs into the bot
- User Submits a survey
  - Sub-flow
    - User logs into the bot and submits questions
    - Prompt comes up to select times and timeslots on when to meet with their project manager
    - From there, the user can book a potential time slot for additional direction
  - Alternate
    - Empty time slot list



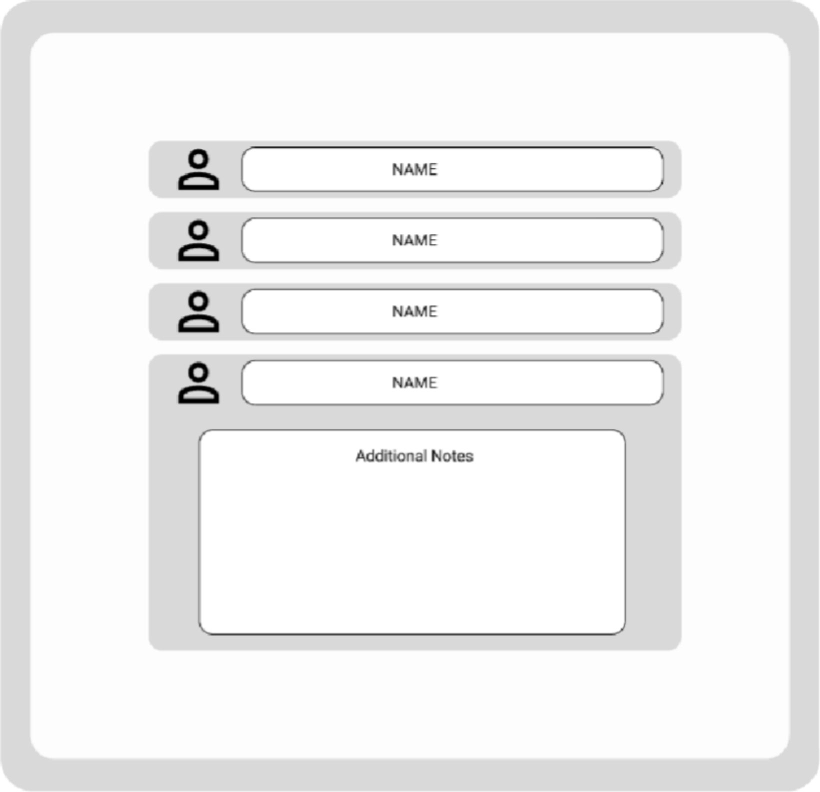
Mockup



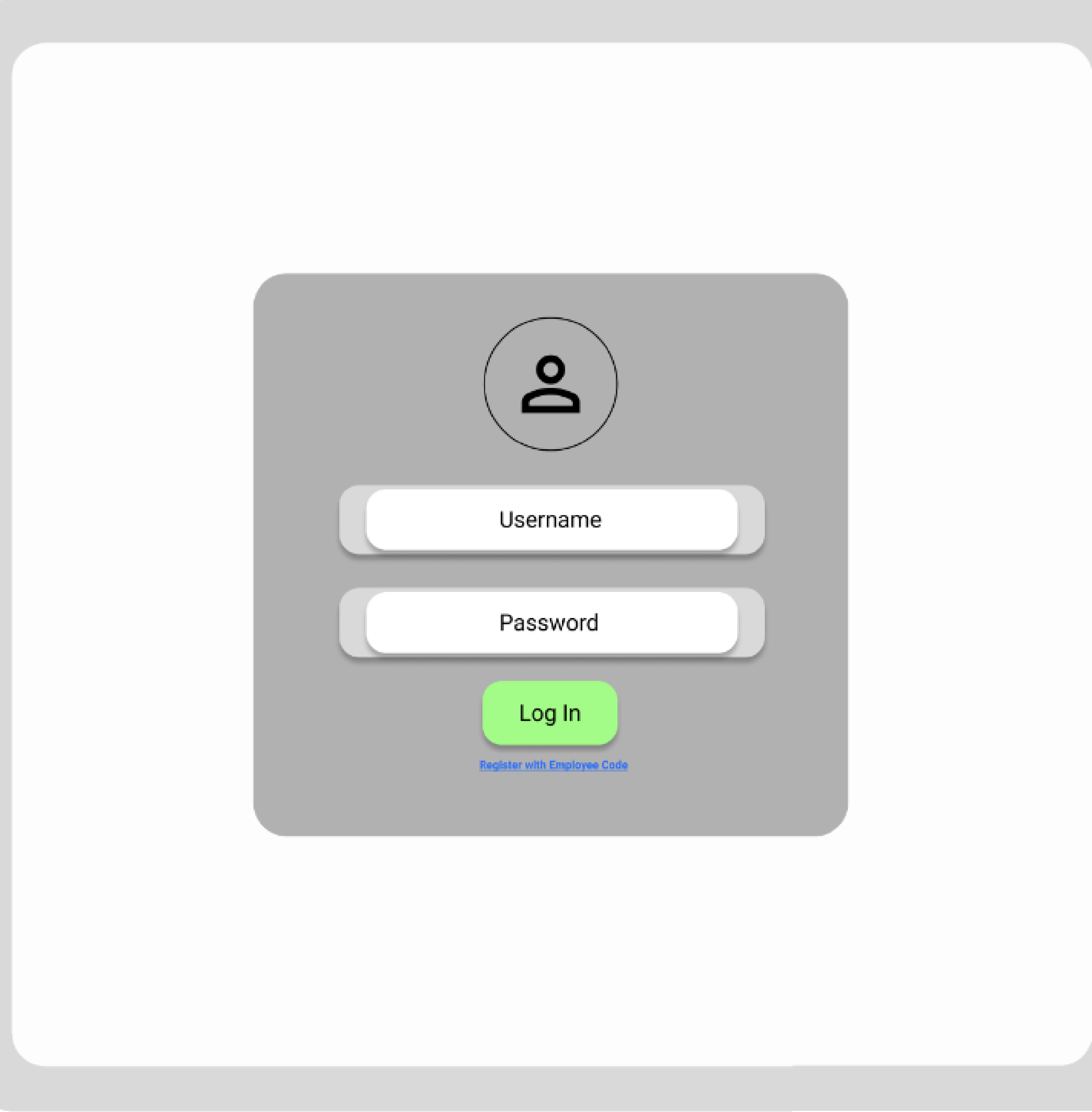
How Comfortable are you with the tickets you have been assigned?

Do you know what the significance of what you've been assigned?

Do you know everything you need to, to complete the tickets?



MOCKUP 2

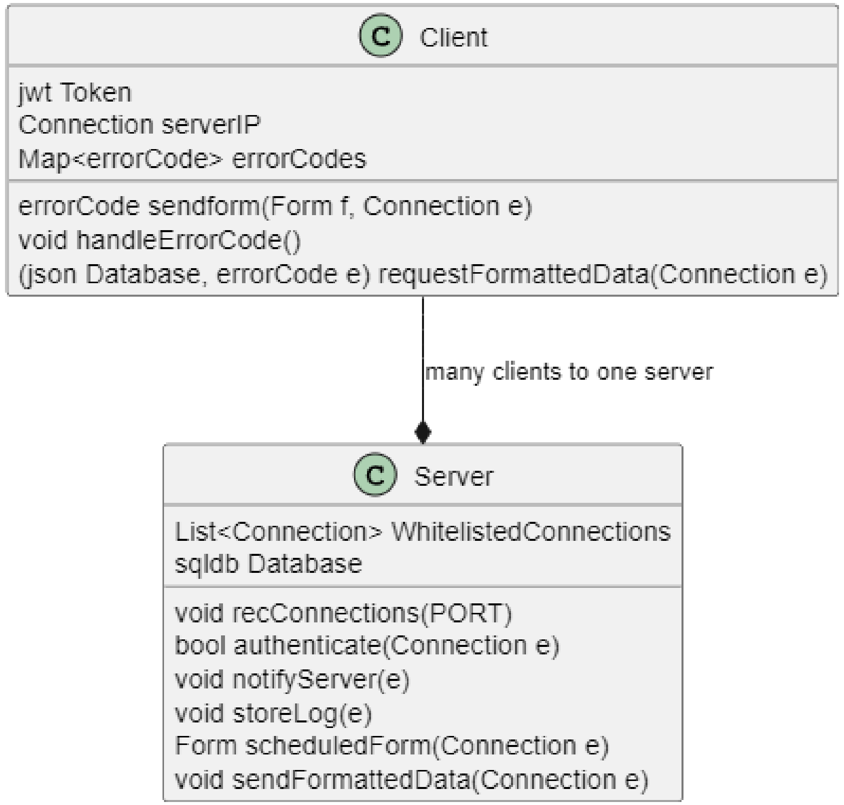


Related Work

- Jira
- GitHub
- Slack
- Teams

Three concepts described in class

- UML
  - Stakeholders
  - Coherence
- SE Process
  - Waterfall
    - Consider the scope and constraints of the project
- Event-Driven Architecture
  - Asynchronous system is efficient
  - Surveys can be submitted at any time
  - Important to have a structured design process


C Client
jwt Token
Connection serverIP
Map<errorCode> errorCodes
errorCode sendform(Form f, Connection e)
void handleErrorCode()
(json Database, errorCode e) requestFormattedData(Connection e)
many clients to one server
C Server
List<Connection> WhitelistedConnections
sqldb Database
void recConnections(PORT)
bool authenticate(Connection e)
void notifyServer(e)
void storeLog(e)
Form scheduledForm(Connection e)
void sendFormattedData(Connection e)

Future Work

- Involving AI to analyze submissions
  - Convert employee confidence into metrics that guide how company resources should be allocated
- Internationalization of the app
  - Support for different languages, regions
- Specialization of the bot for different companies
  - Tailoring the bot to fit business needs
    - Front end, back-end bots
- LLM trained on Company data to provide feedback
  - Replace the need for meeting with tailored responses and guidelines

Works Cited

- [1] W. Scacchi, “Understanding Software Productivity,” ics.uci.edu, https://ics.uci.edu/~wscacchi/Papers/Vintage/Software_Productivity.html.