Team 2

Garrett Davidson, Noah Maxey, Nate Ohlson, Riley Shaw

Problem Statement

Our team previously built an iOS app which relied on a backend written by a previous team member. This app is meant to facilitate learning new languages through talking to native speakers. This server has many limitations, including platform compatibility, speed, and scalability. We would like to scrap that server and start over using different technologies to address these issues. We will not be reusing any of the code because we are transitioning from a Dropwizard based Java server to a fully Swift server. We do not intend to use the app client for any part of this course, the client will help guide the high level design of the new server. Because we do not plan to use the client for any part of this project, we will only be submitting tests for the server. Creating this new server will also allow us to design from the ground up with testing in mind to prevent regressions as we add more features in the future.

Project Objectives

We will build a backend server that will:

A. Have a smaller memory footprint
B. Share core technologies with the client
C. Be deployable on our own servers
D. Run more efficiently for a better user experience
E. Allow for simpler testing and maintenance
F. Be able to scale to support a large number of users

Previous server functionality:

A. Send and receive text messages between other users
B. Create/Update user profile
C. Send and receive multimedia messages

Stakeholders

A. Users: The developers of this project who want a testable and easy to maintain server to support our app.
B. Software Developers: Garrett Davidson, Noah Maxey, Nate Ohlson, Riley Shaw
C. Development Managers: Garrett Davidson
D. Project Owner: Team 2

Deliverables

We plan to create a server with the following features:

A. Receive and store multilingual Unicode compliant messages for the conversations between users.
B. Receive and store picture and audio messages securely and efficiently
C. Adopt Apple Push Notification Service for notifying users of messages and other app relevant information.
D. Allows user registration through Google and Facebook authentication APIs
E. Design a user matching algorithm to pair users who are likely to want to communicate with each other.
F. Adopt Microsoft's translation API for translating user messages between several languages
G. Create store/edit/syncing flash cards from client to server, allowing users to manipulate multiple card sets for learning new words.

This server will be written entirely in Swift and compiled for Linux.
For long-term storage, we will be using a sql database.
For ephemeral storage, we will be using native Swift data structures and storage.