Team FlaskChained
SoftDev1 pd<1>
P2 -- The End
2020-01-05

FINAL DESIGN DOCUMENT

Jesse Hall (PM)
Ayham Alnasser
Hong Wei Chen
Ivan Galakhov

PROGRAM OVERVIEW:

The classic logic game, Sudoku, in a portable website built with (way too much) JavaScript. The puzzles are automatically generated in accordance with difficulty settings: easy, medium, hard, insane. Users can create an account to save finished puzzles and times or can play without an account. A set of leaderboards is displayed on another page, ranking the users by the amount of time it took for them to finish the puzzle, differentiated by the difficulty setting.

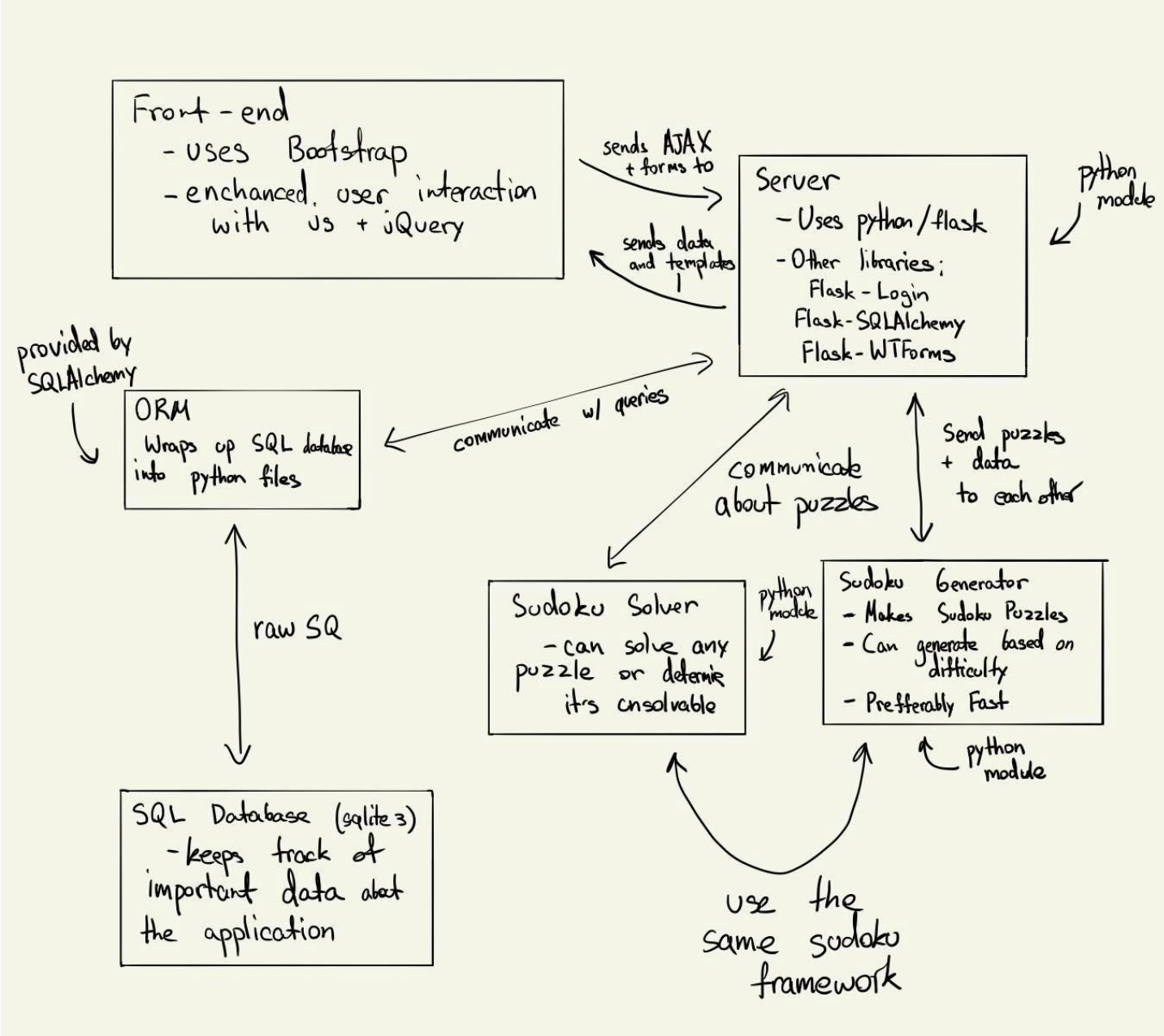
PROGRAM COMPONENTS:

1. **Sudoku Solver/Generator**
   The solver initially creates a unique pseudorandom board that is a legal board. Then, random indices are removed from this legal board, with the amount of removable inversely proportional to the level of the board, e.g. Insane has the most amount of removed indices, and easy has the least amount of removed indices. Then, the program runs the solver again, to ensure that the removed boxes create only ONE legal solution. As a result of the fact that P = NP is not proven yet, it is not possible to create a board with only one legal solution yet.
2. **Front-end sudoku solver**
   The front end sudoku solver provided to the user is the ***capstone*** of this project, and what probably took the longest time to do. Written in way more javascript than an average Stuy student should have to write, the sudoku solver allows the user to modify the presented sudoku with their number keys to place numbers, and shift+number keys to place pencil marks. The sudoku solver also asynchronously communicates with the server to allow the user to refresh their page and not have their progress be lost. Once the user is done, the sudoku solver can pull its own data, and check with the server that the puzzle was solved orr
3. **Database**
   The database uses an Object Relational Mapping (ORM) in order to relate the user ids with the sudoku boards that they have completed, and the save states of each of the users' boards as well. The ORM also uses this relation between boards and to update the leaderboard with the statistics.
4. **Server**
   The server is written in python (flask) and serves the pages to the user. It effectively connects the three components described above together.

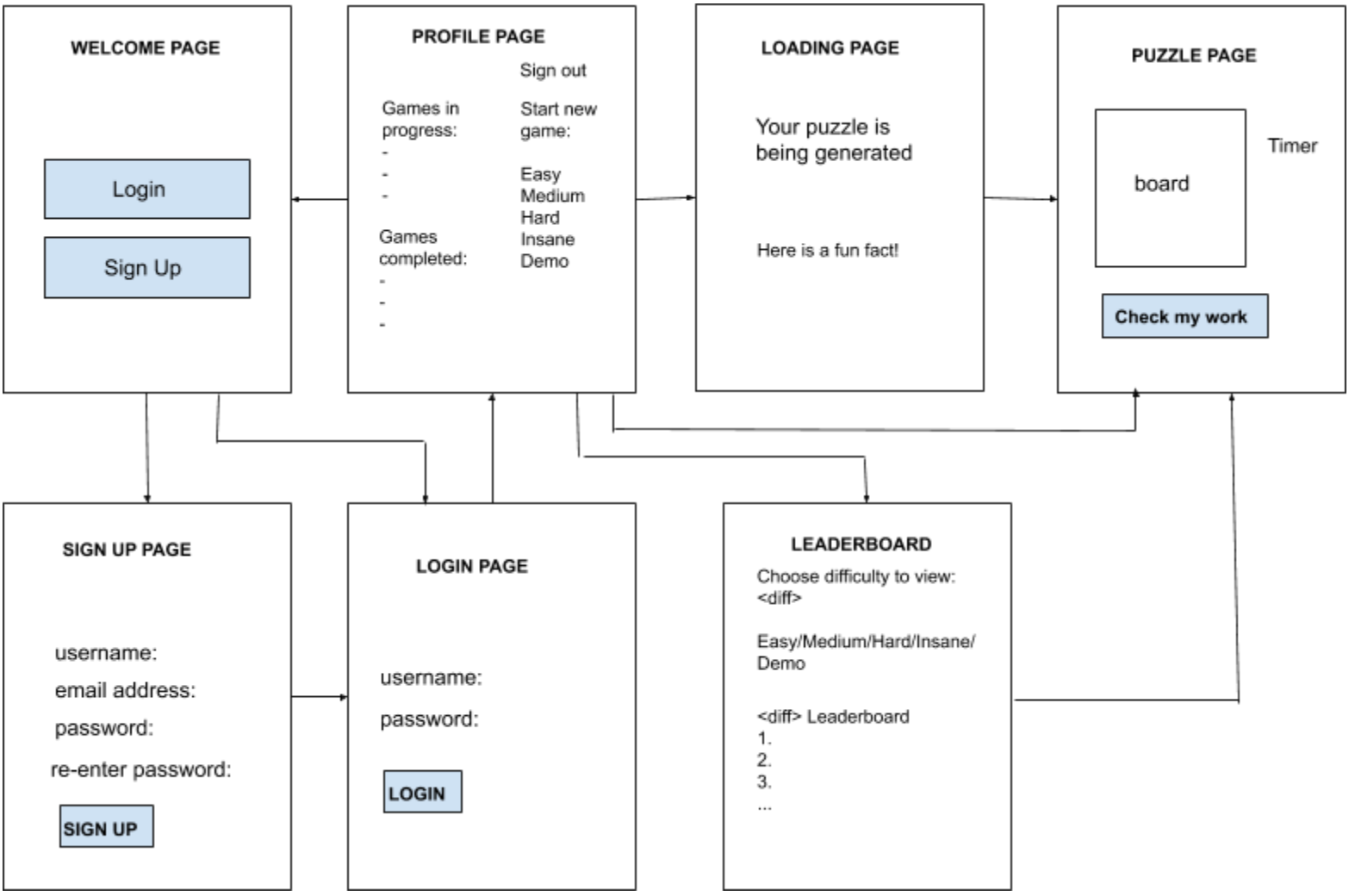
TASKS AND ROLES

1. **Frontend Dev including JS (Ivan)**
2. **DB Management and Design (Ivan)**
3. **User Accounts and log in (Ivan)**
4. **Sudoku Solver and Generator (Ayham)**
5. **Frontend Design including CSS (Hong Wei and Ayham)**
6. **Leaderboards (Hong Wei)**

COMPONENT MAP



SITE MAP



DATABASE MAP