Design Document for The Gameday Companion

Group 1_CW_5
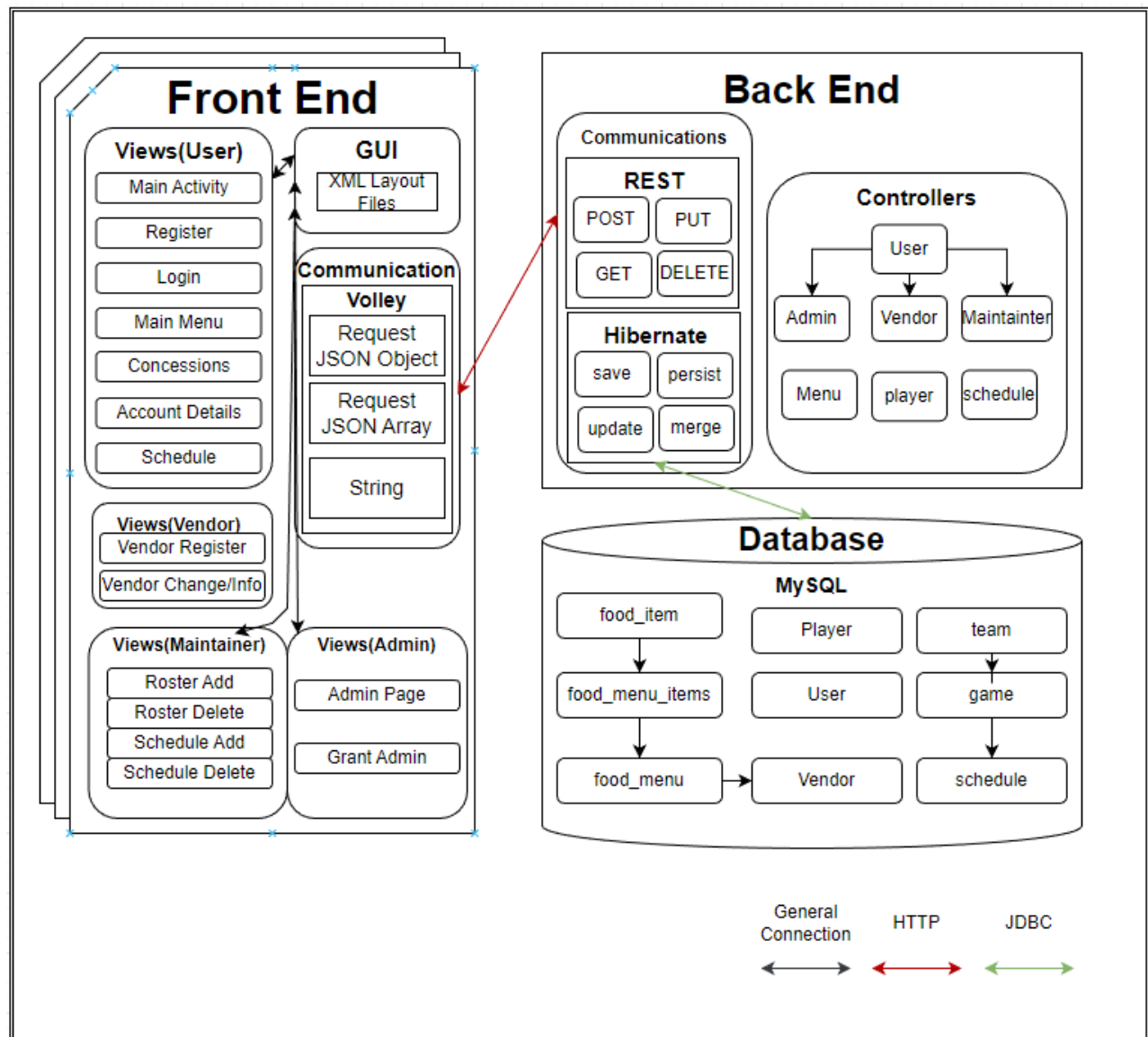
Bryan: 25.1% contribution

Camden: 25.1% contribution

Tyler: 24.7% contribution

Micheal: 25.1% contribution

Block Diagram

Front End

Views(User)
- Main Activity
- Register
- Login
- Main Menu
- Concessions
- Account Details
- Schedule

Views(Vendor)
- Vendor Register
- Vendor Change/Info

Views(Maintainer)
- Roster Add
- Roster Delete
- Schedule Add
- Schedule Delete

Views(Admin)
- Admin Page
- Grant Admin

GUI
- XML Layout Files

Communication
Volley
- Request JSON Object
- Request JSON Array
- String

Back End

Communications

REST
- POST
- PUT
- GET
- DELETE

Hibernate
- save
- persist
- update
- merge

Controllers
- User
  - Admin
  - Vendor
  - Maintainer
- Menu
- player
- schedule

Database
MySQL
- food_item → food_menu_items → food_menu → Vendor
- Player
- User
- team → game → schedule

General Connection

HTTP

JDBC

Frontend

Register(All Users)-Will create an account for a user with values of username and password POSTed.

Login(All Users)-Users can use this page to login to their already created account

Main Menu(All Users)- Users will use this page to navigate to other parts of the app.

Concessions(All Users)-Will show a list of all vendors, and when clicked, it will show that vendor's menu.

Account Details(All Users)- Will allow users to view their details and change their password

Schedule(All Users)- Will show all games scheduled.

Vendor Views(Vendor)- Vendors can use these views to edit and add things to their store page

Maintainer Views(Maintainer) - Maintainers will be able to edit the various views users can see

Admin Views(Admin)- Admin will be able to change the roles of all users here

Backend

FoodMenu-FoodItem relationship- We use two tables to store menus as well as items separately, so they are not tied together. Due to wanting the user to have the ability to change menus quickly, as well as add to menus, we made items individually from menus. This allows each vendor to have multiple menus and each menu to have multiple items and each item can have multiple menus.

Vendor-User Relationship- For our app, we have our vendors and our users set separately. Because a "vendor" is considered an individual food location, so a user (such as Levy) may have multiple stand locations, and each vendor has one menu. The menus are best explained in the

FoodMenu-VENDOR relationship- The way that the relations are set up between menus and vendors allows each vendor to switch between different menus. This also allows each menu to have multiple vendors attached to it at the same time in case there was some location that shared its stock but had different names/locations.

Team-Game Relationship -Similar to the food menu relationship, we have a team table and a game table, and we have an intermediate table in the middle to link team addresses to store in games.

game
- gameid INT(11)
- time VARCHAR(255)
- away_team_team_name VARCHAR(255)
- home_team_team_name VARCHAR(255)
- Indexes

hibernate_sequence
- next_val BIGINT(20)

user
- username VARCHAR(255)
- email VARCHAR(255)
- password VARCHAR(255)
- perm_lv VARCHAR(255)
- phone_num VARCHAR(255)
- Indexes

player
- id INT(11)
- first_name VARCHAR(255)
- last_name VARCHAR(255)
- player_num VARCHAR(255)
- Indexes

team
- team_name VARCHAR(255)
- img_path VARCHAR(255)
- losses INT(11)
- rank INT(11)
- team_location VARCHAR(255)
- wins INT(11)
- Indexes

food_menu
- menu_id INT(11)
- menu_desc VARCHAR(255)
- menu_name VARCHAR(255)
- vendor_vendor_id INT(11)
- Indexes

food_item
- food_id INT(11)
- cal INT(11)
- name VARCHAR(255)
- price BIGINT(20)
- stock INT(11)
- menu_menu_id INT(11)
- Indexes

food_menu_items
- food_menu_menu_id INT(11)
- items_food_id INT(11)
- Indexes

vendor
- vendor_id INT(11)
- oc BIT(1)
- location VARCHAR(255)
- name VARCHAR(255)
- menu_menu_id INT(11)
- Indexes