Design Document for CyHire

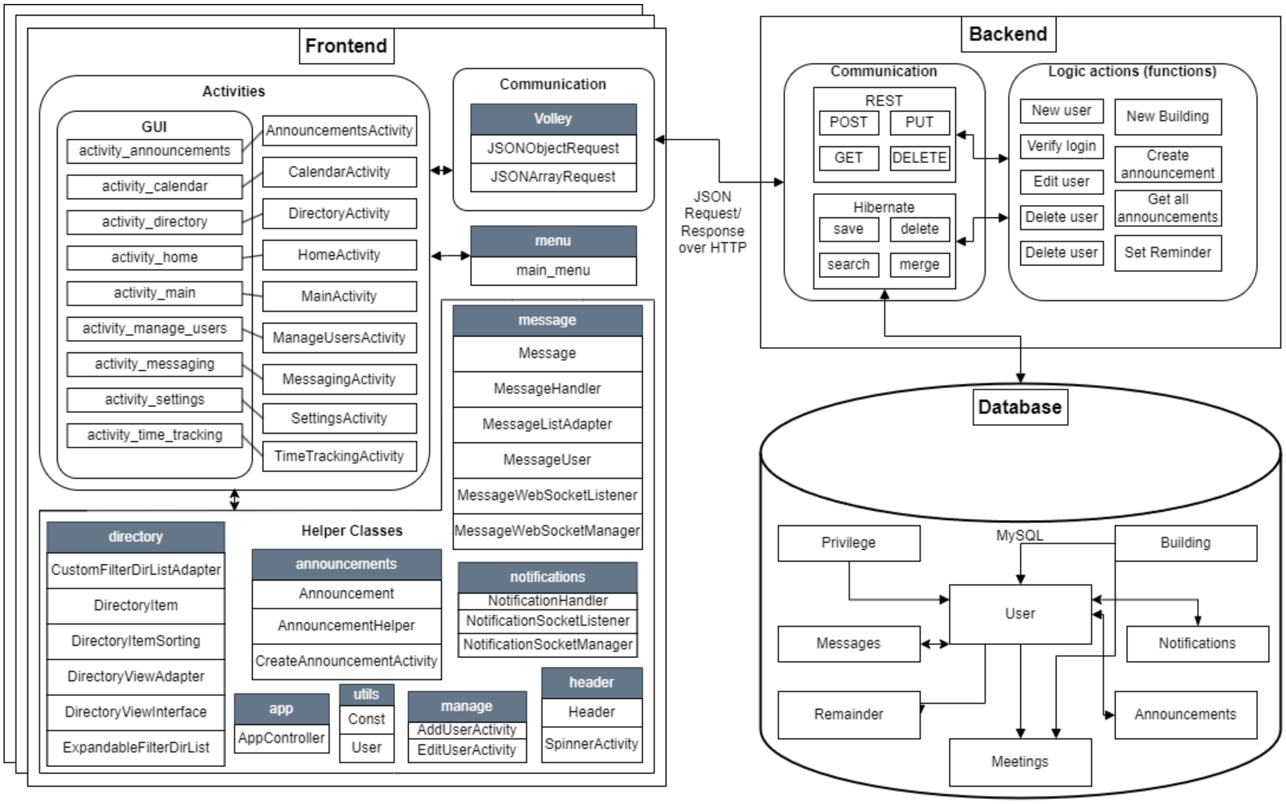
Group IP_104

Yi Ang: 25% contribution

Logan Roe: 25% contribution

Andrew Sand: 25% contribution

Rodrigo Santamaria: 25% contribution


Frontend
Activities
GUI
activity_announcements
activity_calendar
activity_directory
activity_home
activity_main
activity_manage_users
activity_messaging
activity_settings
activity_time_tracking
AnnouncementsActivity
CalendarActivity
DirectoryActivity
HomeActivity
MainActivity
ManageUsersActivity
MessagingActivity
SettingsActivity
TimeTrackingActivity
Communication
Volley
JSONObjectRequest
JSONArrayRequest
menu
main_menu
message
Message
MessageHandler
MessageListAdapter
MessageUser
MessageWebSocketListener
MessageWebSocketManager
Helper Classes
directory
CustomFilterDirListAdapter
DirectoryItem
DirectoryItemSorting
DirectoryViewAdapter
DirectoryViewInterface
ExpandableFilterDirList
announcements
Announcement
AnnouncementHelper
CreateAnnouncementActivity
notifications
NotificationHandler
NotificationSocketListener
NotificationSocketManager
app
AppController
utils
Const
User
manage
AddUserActivity
EditUserActivity
header
Header
SpinnerActivity
JSON Request/ Response over HTTP
Backend
Communication
REST
POST
PUT
GET
DELETE
Hibernate
save
delete
search
merge
Logic actions (functions)
New user
Verify login
Edit user
Delete user
Delete user
New Building
Create announcement
Get all announcements
Set Reminder
Database
MySQL
Privilege
Building
User
Messages
Notifications
Remainder
Announcements
Meetings

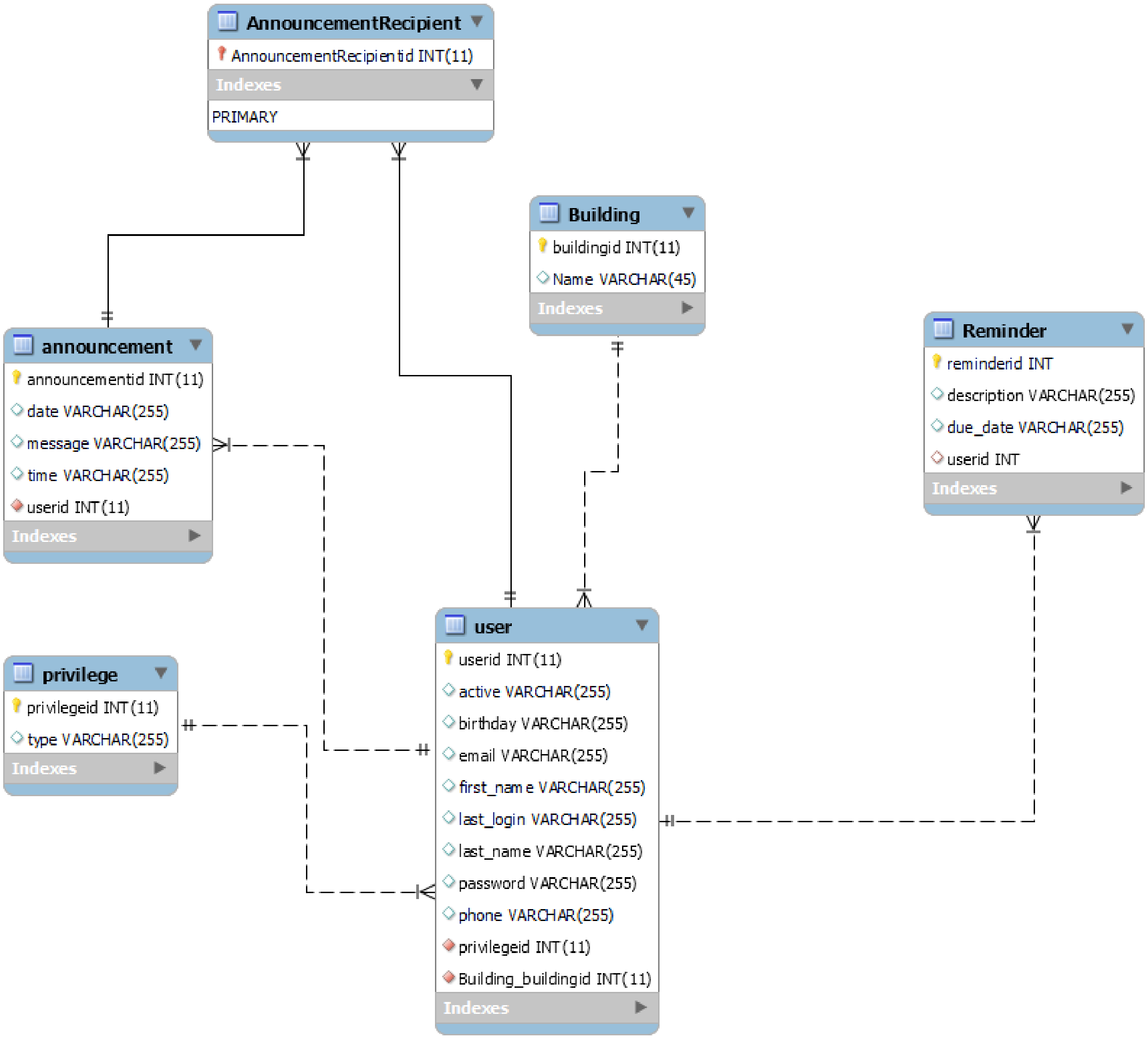
Block Diagram Description

Frontend (Currently Implemented In Main)

- Login
  - Takes input of an email and a password. On the frontend side, it ensures that neither of the entries are empty once the "Login" button is pressed. If either is empty, no request is sent to the server and an error message is displayed. If both have values, it is sent to the backend server and a user object is returned if it is a successful login.
- Home
  - The main landing page when a user logs in. When the activity is loaded, it requests top announcements from the server, which it then displays. Additionally, the home page will display a brief welcome message and date of last login to the user. An error is displayed if the server can not be reached.
- Announcements
  - Upon loading, the announcements page sends a JSON request to the backend to receive all of the announcements. The announcements are displayed in a list that can be scrolled through, and each announcement shows its title, sender, and message. If the server can not be reached, an error message is displayed instead.
  - Additionally, if the logged in user is a building or system administrator, there will be the option to go to a separate page to add a notification or announcement.
  - Add Announcements or Notifications
    - On this subpage, the user can cancel to return to the announcements page or type in a message and click the send button to send the text as a new announcement to the server.
    - Additionally, the user can click a checkbox instead before pressing the send button to send the typed text as a global notification instead, which will show up on all currently logged in users' screens as a popup.
    - For both the announcements and notifications, an error will display if there was a problem with sending the request to the server.
- Employee Directory
  - Used to search for different users with various filters and sorting so that someone can find an employee's email, phone number, or other information about them.
    - Searching can be done via any combination of first and last name or email.
    - Sorting is done via alphabetical or reverse alphabetical, based on last name.
    - Filtering can be done by building, job level, or a combination thereof.
  - Displays results as an object, in card view, with profile picture, name, email, phone number, building, and job level.
- Messaging
  - The logged in user can directly text message any other currently logged in user using this page. The center portion of the page depicts a chat log with the currently selected user to message with a full chat history and timestamps. A drawer menu allows the user to view and select previously messaged users. There is also a plus button in the drawer menu, which will bring up a search bar that allows the logged in user to search for and begin direct messaging another user in the system.
- Manage Users
  - Only available to users who have the privilege level of a system or building admin. This page allows the user to search for any user in the database, edit a user's information, add a new user, or delete a user from the database. An error is displayed if any of the alteration operations fails on either the frontend or backend.
  - Add User
    - The administrator can add a new user by inputting a first name, last name, email address, phone number, privilege level, password, etc.
  - Edit User
    - The administrator can edit any of the chosen user's information in addition to creating a temporary password that the user can use to log into their account.

Backend

- Communication: The backend portion of our project uses request methods (CRUD) to create, read, update and delete the data in our database.
  - Post: Send data to the server to create a resource in the database.
  - Get: Request data from a specified resource in the database.
  - Put: Send data to the server to update a resource in the database.
  - Delete: Deletes a specified resource from the database.
- Controllers: Controllers include the request mappings for communication between the frontend and the database
  - User: Contains mappings to create a new user, lists all users in the database, search for a specific user based on first name and last name, post login request to verify user information for login purposes, deletes a user based on userid and edits user information in the database.
  - Privilege: This has a one-to-many relationship with user, it consists of privileges such as system admin, building admin and regular user. Mappings create a new privilege and lists all the privileges
  - Announcement: This has a many-to-one relationship with user. Mappings create a new announcement list all announcements.
  - Reminder: This has a many-to-one relationship with user. Mappings create a new reminder and lists all the reminders.
  - Building: This has a one-to-many relationship with user. Mappings create a new building and lists all the buildings.
  - Hour: This has a many-to-one relationship with user and hourType. Mappings create a new hour and lists all the hours.
  - Hour Type: Mappings create a new HourType and list all hour types.
  - Message: Controller for storing web socket messages, used to list user chat history.



Swagger API URL: http://coms-309-010.class.las.iastate.edu:8080/swagger-ui/#/