MeetUP

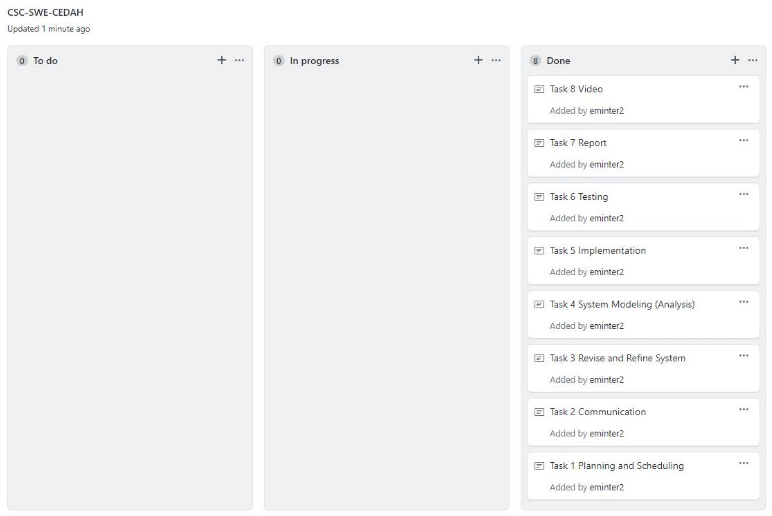
Spring-2020

Team: CEDAH
Ethan Minter
Nghi Tran
Harry Park
Daniel Busby
Connor Lacy

Planning and Scheduling

Team Member	Email	Task	Duration	Prereq.	Due Date
Harry Park	hpark44@student.gsu.edu	6	3 hours	Task 5	3/3/20
Daniel Busby	dbusby1@student.gsu.edu	5	5 hours	None	3/2/20
Ethan Minter (Coordinator)	clacy2@student.gsu.edu	1,2,3,4,7	3 hours	None	3/3/20
Connor Lacy	eminter2@student.gsu.edu	5	5 hours	Database	3/3/20
Nghi Tran	ntran52@student.gsu.edu	4,8	4 hours	None	3/3/20

Everyone did their job as assigned.



CSC-SWE-CEDAH
Updated 1 minute ago

0 To do

0 In progress

8 Done

Task 8 Video
Added by eminter2

Task 7 Report
Added by eminter2

Task 6 Testing
Added by eminter2

Task 5 Implementation
Added by eminter2

Task 4 System Modeling (Analysis)
Added by eminter2

Task 3 Revise and Refine System
Added by eminter2

Task 2 Communication
Added by eminter2

Task 1 Planning and Scheduling
Added by eminter2

Teamwork Basics

o Ground Rules: Norms 1 to norms 5

Work Norms: All of the assignments are distributed upon the interest and the availability of time. The group coordinator will set the deadlines for everyone. When someone doesn't follow through on their commitment or people have different opinions about the quality of work, all group members will collaborate to fix the problem. Different work habits are not a big problem unless people can get the assignments done in time. We agreed upon not procrastinating this project as a last-minute assignment.

Facilitator Norms: Our group will have a facilitator. The group coordinator will be the facilitator. As the group coordinator is going to be rotated for each assignment, every member will have a chance to learn new things as a facilitator. The responsibility of the facilitator is (1) to encourage the members to give their hundred percent of the work to the assignment, (2) help them to figure out an alternative if any problems occur, and (3) make sure they finished the tasks on time.
Communication Norms: The communications are being done via Discord app so everyone could see the messages and reply instantly.

Communication Norms: The communications are being done via Discord app, so everyone could check the messages and reply instantly.

Meeting Norms: All the group members have different availabilities with work schedules and other classes. The group coordinator is responsible for deciding the meeting time, but every person should agree on the decision. Everyone has his preference about the meeting time and place. The best place for the meeting would be the library rooms. If any members are late or absent to a meeting, we are going to post the things discussed during the meeting on the Discord app so they would not be left behind during the assignment. If they miss several meetings, we would have to talk with the instructor to resolve the issue.

Consideration Norms: Most of the time, the meeting will take place at library rooms. So, people might bring food based on library policy. If one person dominates the discussion, others need to make sure that he listens to others' ideas unless he is making decisions as a coordinator. If someone is not comfortable with the norms, they can be changed by discussing with the team members and coming up with an agreement.

o Hints for Handling Difficult Behavior

If a person is overly talkative, others could try to discourage them from dominating the discussion subtly. It should be done gently so the person might not lose their enthusiasm for the project. If the person insists on their behavior, the group coordinator should have a private conversation with them to encourage the person to hear others' opinions.

If a person is having difficulties communicating with others, others could ask them about their opinion, offer any help they might need, and encourage him to help others. Asking questions to break the ice and get familiar will make the person feel comfortable and be able to adjust to the group environment. Another way is to start talking with the person in Discord, as he may prefer online discussion over the face-to-face conversation.

One of the group members has experienced this during the Web Programming class project. When they first formed a group with another classmate, they seemed uncomfortable with discussing the project in person and remained quiet and non-assertive. For introverts, it is natural for them to take their time before saying something instead of thinking out loud. They might seem reserved because of this behavior, but the words they share can have much more thoughts behind them. So, their strength could be stood out within an online discussion. While having a conversation online, they were able to see that this person has a strong knowledge and initiative to lead the project.

If a person would like to give feedback to others, it can positively affect the performance and productivity of a group. However, everyone needs to be careful when criticizing others as it can hurt not only the person being criticized by making him discouraged but also the whole group by dragging out the discussion.

If a person keeps complaining, others should first try to listen to them to see if their complaints are legitimate. If so, the whole group should start a discussion to resolve the issue. Otherwise, other members should remind them part of the work is learning how to solve problems and motivate them to see the bigger picture, setting aside the small conflicts and disagreements.

o Hints for Handling Group Problems

When the groups have just been formed and members are getting to know each other, the productivity might not be in its full potential. In this phase, setting up a common goal and creating a list of tasks to accomplish the goal will help to unify the team.

If the conversation is not focused on the work at hand, the progress of a group can be dragged out. When the discussion goes off at a tangent, at least one of the members should be ready to remind the group to go back to the main topic.

It would not be a great idea to make a decision quickly and put pressure on others to follow. Understanding is a critical factor in making a decision. Everyone should propose their ideas in a discussion, and the group should decide as a whole.

We should try to cooperate with each other, setting a common goal that everyone can agree on. If the group is not able to reach a consensus for a particular issue, it will have to decide by majority rule unless there is an alternative that can make everyone satisfied.

A conflict between the members might slow down the progress. As it is difficult to move forward without resolving the conflict, they will have to discuss the problem first. They should try to listen to what the other person has to say and think of the problem in other's side.

Every person in the group matters. It is not acceptable to ignore or ridicule another member during the project. In the workplace, it is necessary to know how to deal with people that we are not comfortable with.

There is always a group member who is not willing to cooperate, does not complete their tasks in time, or is absent to the scheduled meetings. The group coordinator should be ready to talk to the uncooperative person, explaining how their behavior negatively affects the whole group. If the person is not willing to change, it would be necessary to talk with the instructor to address the problem.

One of the group members has experienced this during the Mobile App Development class project. Their classmate was responsible for implementing a couple of fragments for the Android application. However, they kept creating unreasonable excuses and had no contribution to the project. As their behavior has not changed after several reminders, other members had to speak to the instructor about the problem. They ended up having a significant penalty for the project grade. It would be ideal if we could convince the person to be more cooperative. However, it is sometimes necessary to blow the whistle on selfish, stubborn behavior.

Problem Statement

Our product is a mobile application that helps users schedule meetings. It is designed for teams and other small groups in both business and casual settings. It streamlines the scheduling process by handling the recognition of overlapping available times, and eases the decision making process. The original alternative to our product was emailing everyone involved and trying to figure out availability through back and forth communication. This process is inefficient because each person generally only knows their schedule. Our app will present available meeting times without needing discussion and debating.

The objective of our project is to allow users to make a group in the app, and then send notifications to group members stating their desire to schedule a meeting within the next week. Then, users indicate their availability at various times on each day within the period. Users can designate a time as unavailable, available, and partial availability (this can include an undesired time or a virtual presence in an in-person meeting). After the app determines the overlapping times, ranks them by percentage of participants that are fully available, then by partial availability. Users then vote for a time. We will keep the scope to scheduling a meeting within a one week period because setting availability for more than 7 days would be tedious, and schedules are often much more variable outside of that time.

Our main competitor is Doodle. Doodle's approach is to allow hosts of events to send times to group members and have them take a poll on which they prefer. Our approach will be to have the app find availabilities for the user, which will provide options that a host could possibly miss. In addition, we will have our app adapt to emergency schedule changes automatically. When a user changes availability on an already voted on event, a message will be sent to the group leader asking if the group should vote again. Then, the app repopulates the list with the overlapping times, and the group decides another time. Another feature that will differentiate our product will be an agenda feature that implements image recognition technology to translate handwritten notes into digital notes for the meeting.

Requirements

The requirements were elicited through two main methods. Firstly, we faced a personal need for a mobile scheduling application and thought about what features would be most beneficial for us. Secondly, we looked at an already existing scheduling application, Doodle, and figured out what it's functionalities were and what features our product would also need.

User Functional Requirements

1. The group leader must be able to invite other users to a meeting.
2. Users must be able to set availabilities.
3. Users must be able to designate partial availabilities, and give a reason why.
4. The users must be able to vote on overlapping times.
5. Users then vote on the times in order to determine the best time for everyone.
6. The app should repopulate the list of times if a schedule changes.
7. The app should be able to convert handwritten notes to text notes for the meeting.

System Requirements

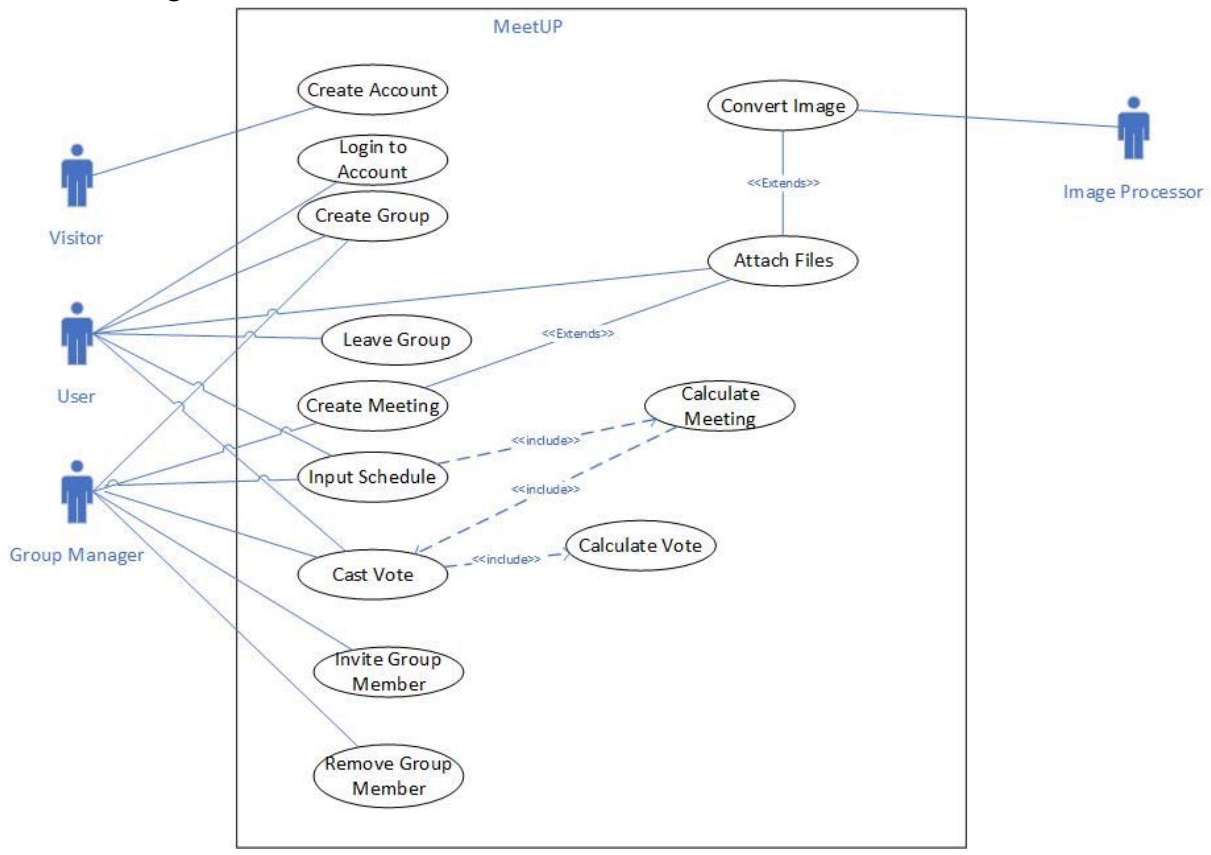
Functional Requirements:

1. Creation of user accounts (from user via web page)
2. Creation of a group of users. (from use via web page)
3. Creation of a new meeting including a list of attendees within a group. (from user in group, via web page)
4. Accept schedules of users (from user via web page)
5. Calculates meeting times from input schedules (at time of meeting creation)
6. Create a vote for an appointment time and date. (at time of meeting creation)
7. Meeting invitations accept attachments (at time of meeting creation)
8. Convert handwriting into text (on request, for meeting attachments)

Non-Functional Requirements:

1. User credentials must be stored securely using hash
2. Each group must have a Manager and hierarchy of inheritance decided by manager.
3. Schedules may need to be confidential depending on type of organization

Use Case Diagram:



Use Case Specifications:

<u>Register Account</u>

Summary: The user creates an account on MeetUP.

Basic Course of Events

1. The user selects the create account option in the menu
2. The user is asked to input a username and password.
3. The system searches the database to confirm that the username and password are unique.
4. If the username is unique, the username is stored in the database.
5. If the password is unique, the password is hashed and then stored in the database.
6. A user id is created for the user.
7. The system outputs a message stating that registration was successful.
8. The user is taken to the login screen.

Alternate Paths: Users can at any time cancel the registration process.

Exception Paths: If the username or password is not unique, the user is asked to input a new username and password.

Preconditions: None

Postconditions: An account has been created by the user.

<u>Login to Account</u>

Summary: The user inputs their username and password and logs in.

Basic Course of Events

1. The user selects the log in option in the menu.
2. The user is asked for their username and password

3. The system checks to see if the username exists.
4. The System checks to see if inputted password matches the password associated with the username.
5. If the username exists and the password is correct, the user is logged in.
6. The user is then taken to the main menu.

Alternate Paths: If the user has just registered, then they will enter the login screen as soon as they do so.
Exception Paths: If the username or password are invalid, then the user is asked to input their login again.
Preconditions: None
Postconditions: The user has logged into their account.

<u>**Create Group**</u>
Summary: The user creates a group and sends invitations to other group members.
Basic Course of Events

1. The user selects the create group option.
2. The user is asked to input a name for their group.
3. The user invites other users by inputting their usernames.
4. The invited users receive the invitation, and either accept or reject the invitation.
5. Once a user accepts the invitation, they are added to the group.
6. The group creator becomes the Group Manager for the group.

Exception Paths: If the user inputs an invalid username in step 3, they are asked to input another username.
Precondition: The user is logged in.
Postcondition: The group is created and members are invited into the group.

<u>**Leave Group**</u>
Summary: A user leaves a joined group.
Basic Course of Events

1. The user selects a joined group.
2. The user selects the leave group option.
3. The user id of the user is no longer associated with the group.

Alternate Paths: The user can cancel leaving the group.
Precondition: The user must be part of a group.
Postcondition: The user is no longer a member of the left group.

<u>**Create Meeting**</u>
Summary: The Group Manager creates a meeting to take place within the next week.
Basic Course of Events

1. The group manager selects the create meeting option.
2. The group manager can set the meeting to repeat next week.
3. The group manager names the meeting and sets the length.
4. Each user receives a notification of the meetings creation.
5. Complete use case input schedule for each user.
6. Complete use case calculate meeting.
7. Complete use case cast vote.

8. Complete use case calculate vote.
9. The meeting time is sent to each user.
10. The group manager can attach files to the meeting, using the Attach Files use case.

Alternate Paths: If the meeting is set to repeat, at the beginning of the next week the process begins at step 3.
Precondition: The user must be the Group Manager of a group.
Postcondition: A meeting is scheduled, and all group members are notified on the date.

<u>Input Schedule</u>
Summary: The user inputs their availability for the given week.
Basic Course of Events

1. The user can indicate that their availability is the same as last week, or they can choose to make a new schedule.
2. If the user has a new schedule, they are asked what days that week they are available.
3. The user then indicates available times by selecting sections of time during that day.
4. The user can select from a list of types of partial availability and indicate sections of time where that applies.
5. Step 3 and 4 are repeated for each available day.
6. The user selects the finish schedule option, and the availabilities are saved.

Alternate Paths: If the user does not want to change their schedule from a previous week, the process ends at step 1. If they wish to modify an existing schedule, the user is instead asked to indicate what day to change availability for in step 2.
Precondition: User must be part of a meeting.
Postcondition: The user's availability is set and ready to meeting time calculation.

<u>Calculate Meeting</u>
Summary: The system takes the schedules of the users within a group and finds the overlapping times.
Basic Course of Events

1. The system analyzes the schedules of the users and finds overlapping complete availability from all users.
2. The system then analyzes the schedules to find overlaps that include partial availability from some users.
3. The system, if overlaps were located, lists the overlaps by percentage of complete availability, then percentage of partial availability.
4. The system then presents users with the list of overlaps to be voted on.

Alternate Paths: If no overlaps can be found, then the group manager and all group members are alerted.
Precondition: A meeting must have been created and all members of the meeting must have a created a schedule.
Postcondition: The group now has a list of meeting times to be voted on.

<u>Calculate Vote</u>
Summary: The system selects the overlap that has the most votes, and reports the winner to the group.
Basic Course of Events

1. Once all users in a group have cast their votes on an overlap, then the system counts the number of votes each option received.
2. The system finds the overlap that received the most votes.
3. The system sends a message to all group members indicating the winning time slot.

Precondition: All users must have voted for a time slot.
Postcondition: The meeting time has been selected.

Invite Group Member
Summary: The group manager invites a new user into the group.
Basic Course of Events:
1. The group manager selects the add member option.
2. The group manager inputs the username of the invitee.
3. The invited user can choose to either accept or reject the invitation.
4. Upon accepting the invitation, the user is now a member of the group.

Alternate Path: The invited user rejects the invitation.
Exception Path: If the username does not exist, then the group manager receives an error that states that the user does not exist.
Precondition: The invitee must not already be a member of the group.
Postcondition: The invitee is now a member of the group.

Cast Vote
Summary: Users and Group manager can vote on the suggested meeting times
Basic Course of Events:
1. A group meeting is proposed by a group manager
2. Scheduling engine builds suggestions for meeting times
3. Notification is distributed to all members of group
4. Group members cast their vote on the suggested meeting times
5. Votes are tallied and used for meeting creation

Alternate Paths:
- User or Group Manager does not cast vote

Exception paths:
- User contact information out of date or unavailable

Precondition:
- Users and group manager are in a group together
- Users have input their schedule for the week

Remove Group Member
Summary: A group member is removed from the group by the Group Manager
Basic Course of Events:
1. A group manager enters a group they manage
2. The group manager removes a group member from the group.
3. The group member removed will no longer receive updates or notifications about future meetings

Alternate Paths:
- A group member may leave the group on their own accord

Exception Paths:
- A group member who is not the group manager attempts to remove another individual

Precondition:
- The member of the group who wishes to remove another member on their behalf is a group manager
- Group manager can only remove individuals from groups they manage

__Attach Files__
Summary: A group member can attach files to meeting
Basic Course of Events:
1. A group member attaches a video, image, or document to the meeting.
2. If it is an image, the user can indicate that there is handwriting to be converted to digital text.
3. If the user wants the image converted, complete use case Convert Image.

Alternate Paths:
- No group members upload attachments to the meeting

Exception Paths:
- Attachment too large to upload
- Broken link or incompatible attachment type

Precondition:
- A meeting exists with members
- Meeting able to accept attachments

__Convert Image__
Summary: Image processor converts image to digital text; attached to meeting description
Basic Course of Events:
1. An image frame
2. The image is processed with NLP techniques
3. Handwritten text is converted to digital language
4. Digital conversion is parsed into meeting body

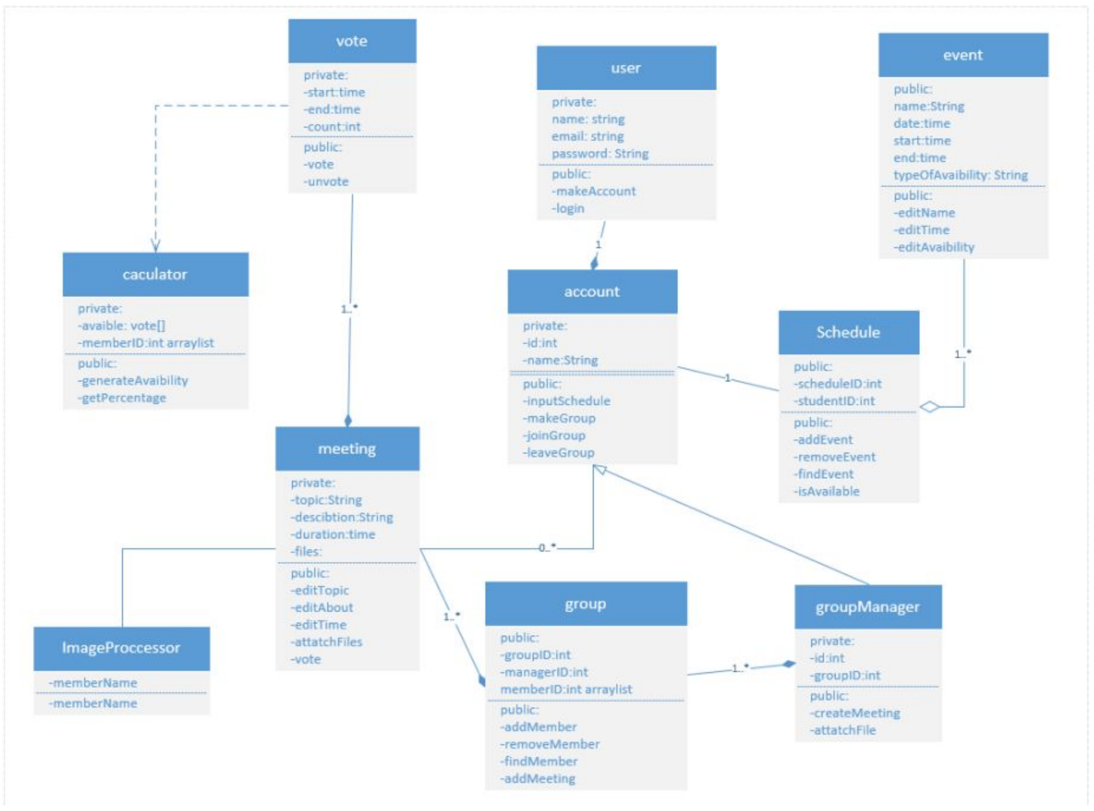
Alternate Paths:
- The camera capture can be cancelled
- Image read from device local storage

Exception paths:
- Camera unable to be detected
- Image processing unable to read handwritten language

Precondition:
- An image is able to be read from a camera or device storage

System Modeling

Class Diagram:



Database Specifications:

Since the project handles structured data, relational database would be ideal for storing and retrieving information with queries. Relational database also allows tiered access in applications. For example, users could view their accounts while managers could both view and make necessary changes. Among a few different SQL-based database management systems, MySQL will be used for the project.

1) Logins

ID	Username	Password
int	char (50)	char (50)
Primary Key		

2) Users

ID	Name	E-mail	Phone	Group_ID
int	char (50)	char (50)	char (50)	int
Primary Key				Foreign Key

3) Groups

ID	Meeting_Day	Meeting_Start_Time	Meeting_End_Time	Manager_User_ID
int	char (50)	time	time	char (50)
Primary Key				Foreign Key

4) User Availabilities

ID	Day	Start_Time	End_Time	User_ID
int	char (50)	time	time	int
Primary Key				Foreign Key

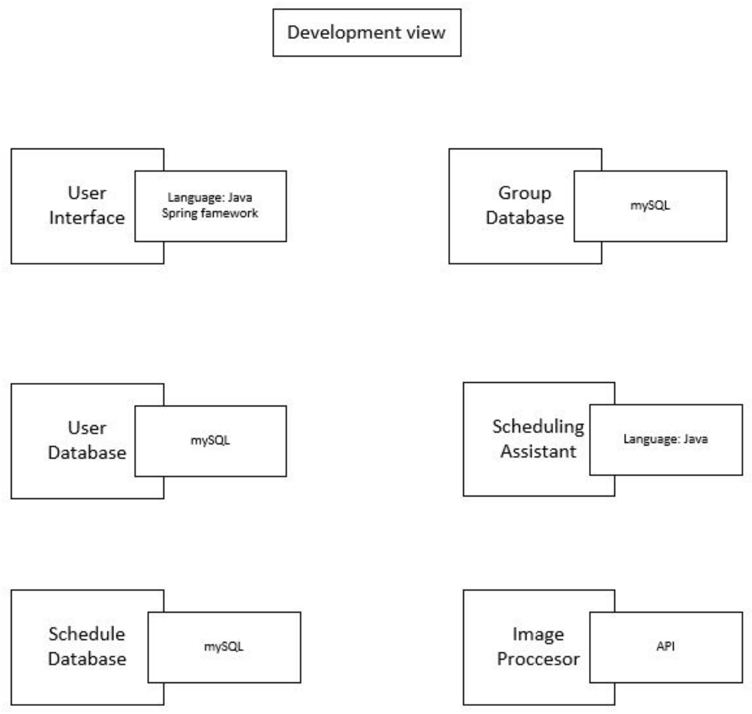
5) Meeting Possibilities

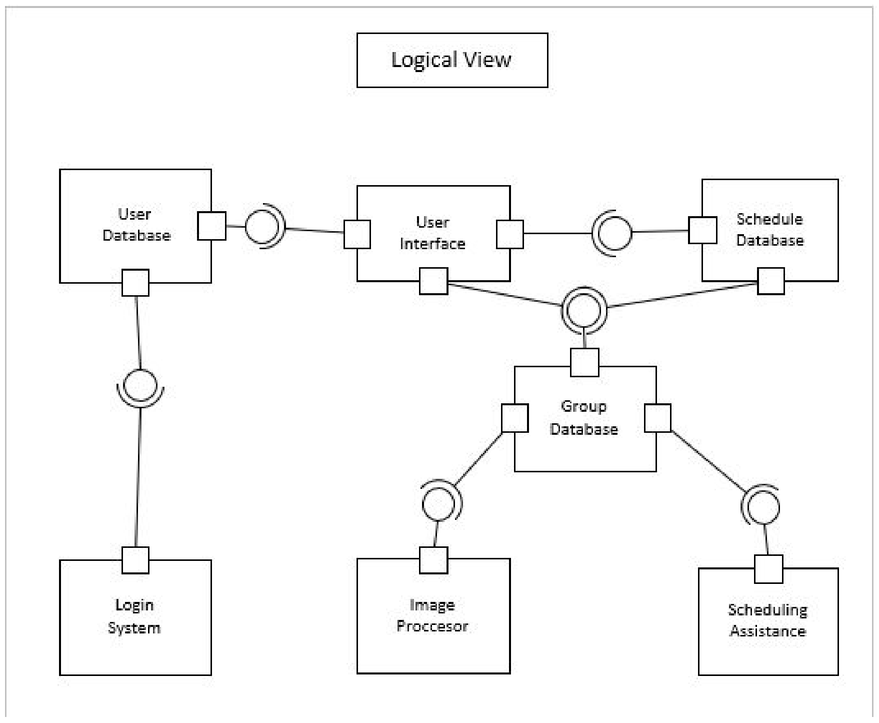
ID	Day	Start_Time	End_Time	Vote_Count	Group_ID
int	char (50)	time	time	int	int
Primary Key					Foreign Key

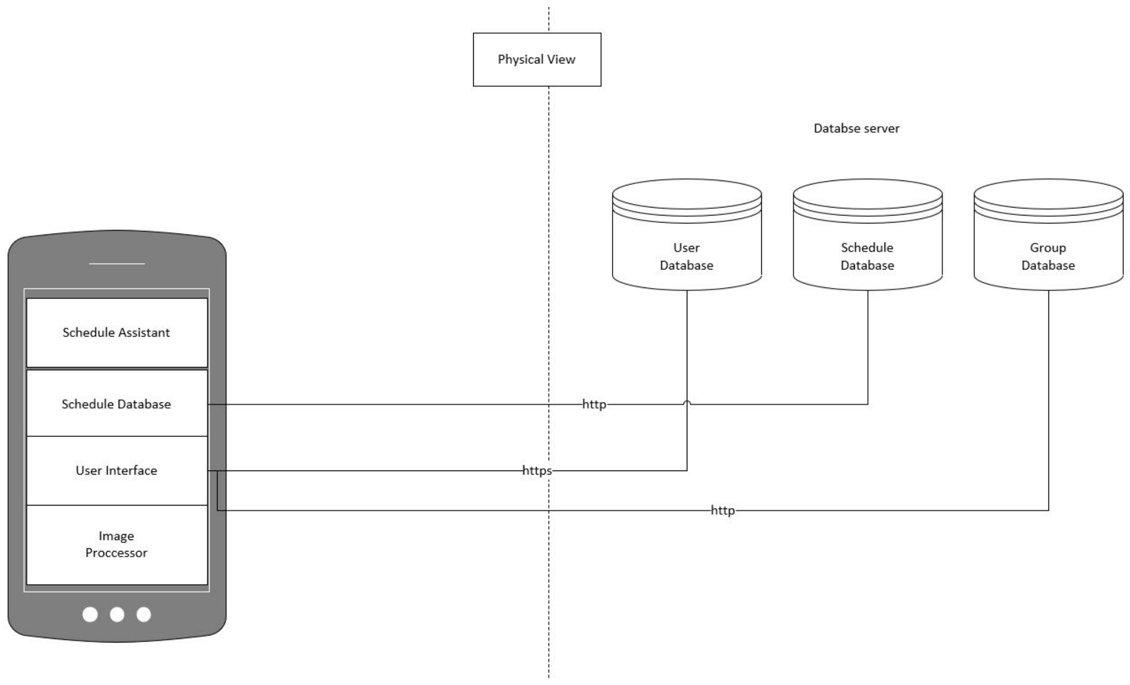
6) Files

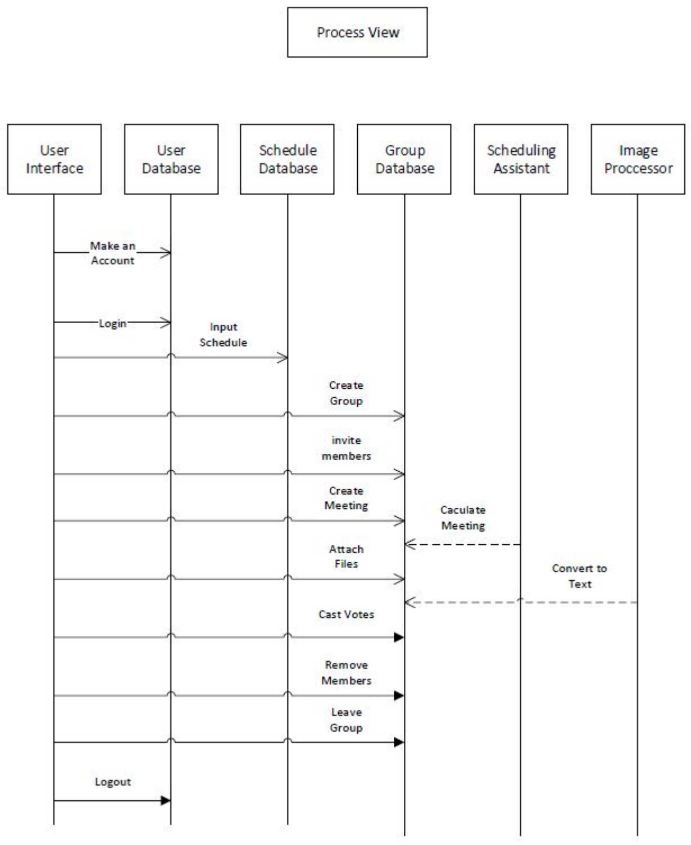
ID	Description	Data	Filename	Filesize	Filetype	Group_ID
int	char (50)	longblob	char (50)	char (50)	char (50)	char (50)
Primary Key						Foreign Key

4+1 Diagram


Development view
User Interface
Language: Java Spring famework
Group Database
mySQL
User Database
mySQL
Scheduling Assistant
Language: Java
Schedule Database
mySQL
Image Proccesor
API

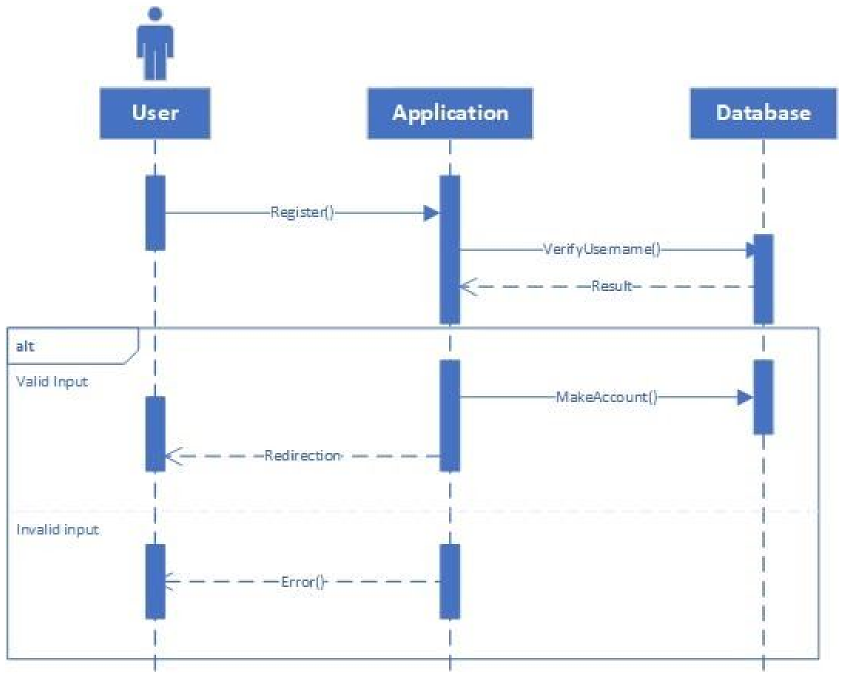

Logical View
User Database
User Interface
Schedule Database
Group Database
Login System
Image Proccesor
Scheduling Assistance


Physical View
Databse server
User Database
Schedule Database
Group Database
Schedule Assistant
Schedule Database
User Interface
Image Proccessor
http
https
http

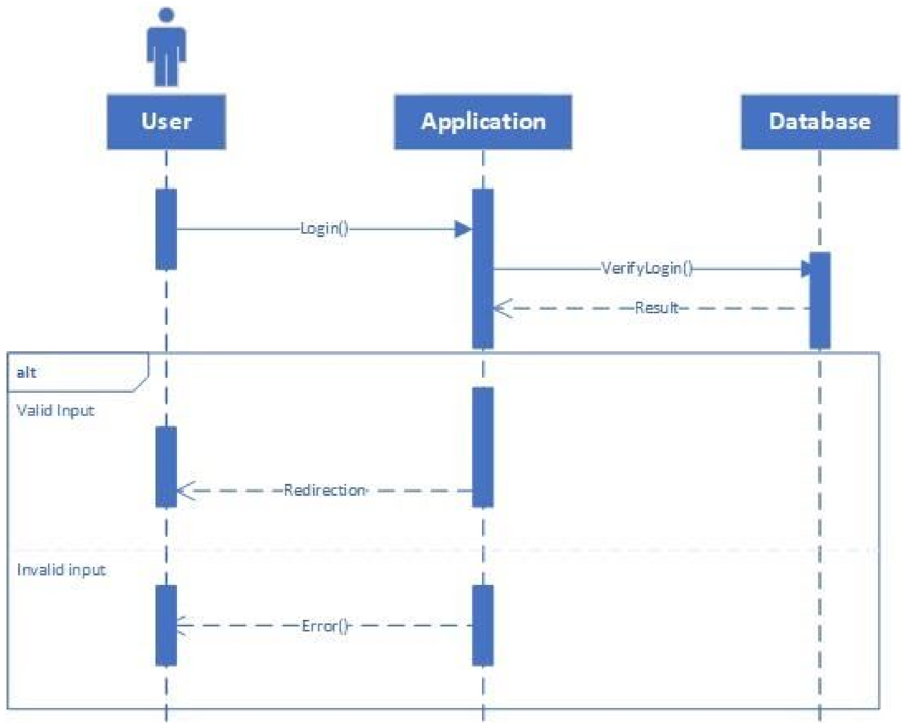

Process View
User Interface
User Database
Schedule Database
Group Database
Scheduling Assistant
Image Proccessor
Make an Account
Login
Input Schedule
Create Group
invite members
Create Meeting
Caculate Meeting
Attach Files
Convert to Text
Cast Votes
Remove Members
Leave Group
Logout

Sequence Diagrams

Use Case: Register

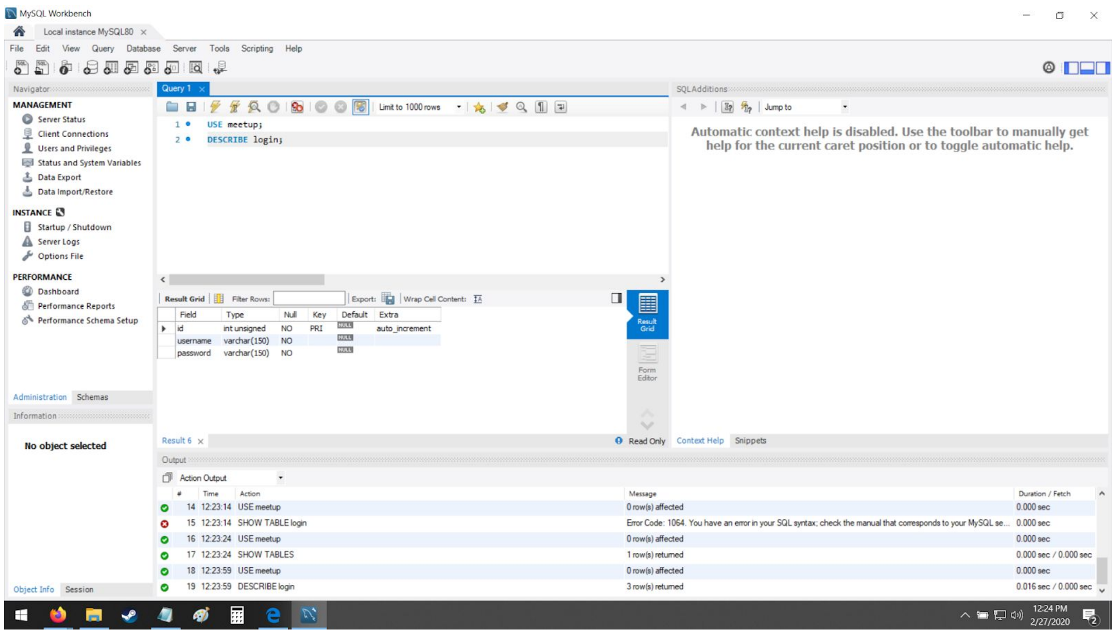


Use Case: Login



Implementation

Database

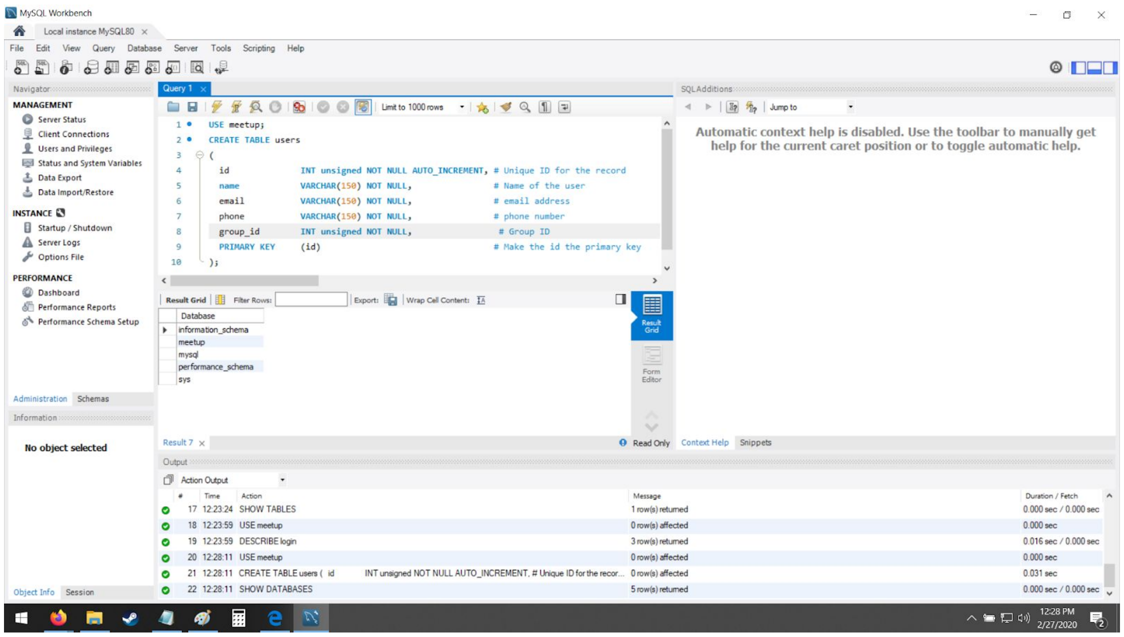


```sql
USE meetup;
DESCRIBE login;
```

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
username	varchar(150)	NO		NULL	
password	varchar(150)	NO		NULL	

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#   Time      Action             Message                                                                                   Duration / Fetch
14  12:23:14  USE meetup         0 row(s) affected                                                                         0.000 sec
15  12:23:14  SHOW TABLE login   Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...   0.000 sec
16  12:23:24  USE meetup         0 row(s) affected                                                                         0.000 sec
17  12:23:24  SHOW TABLES        1 row(s) returned                                                                         0.000 sec / 0.000 sec
18  12:23:59  USE meetup         0 row(s) affected                                                                         0.000 sec
19  12:23:59  DESCRIBE login     3 row(s) returned                                                                         0.016 sec / 0.000 sec
```

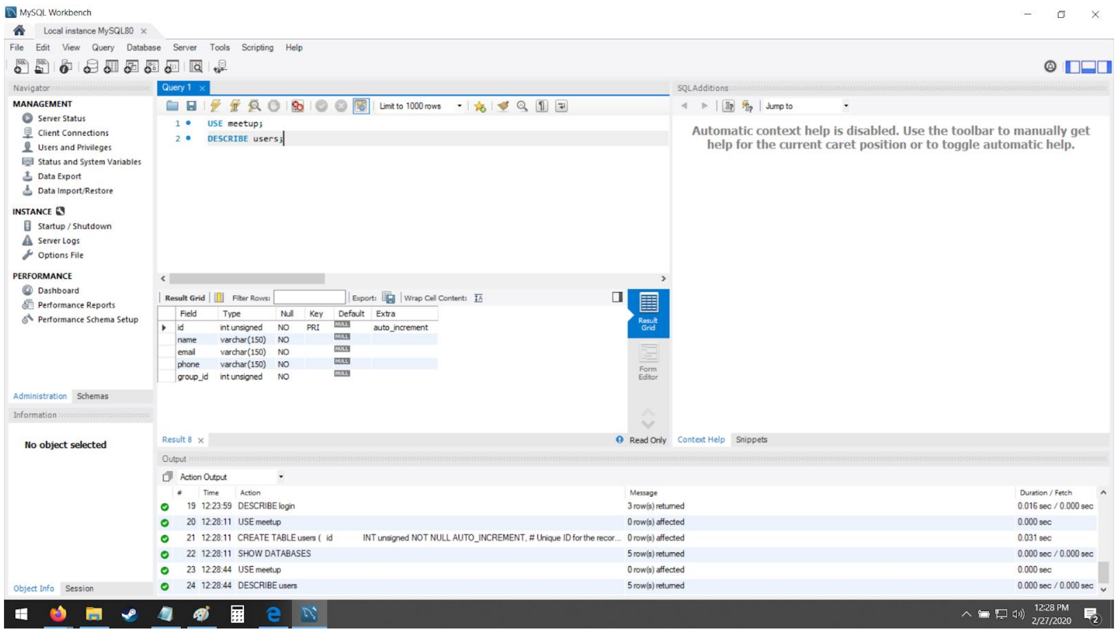


```sql
USE meetup;
CREATE TABLE users
(
  id              INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the record
  name            VARCHAR(150) NOT NULL,                # Name of the user
  email           VARCHAR(150) NOT NULL,                # email address
  phone           VARCHAR(150) NOT NULL,                # phone number
  group_id        INT unsigned NOT NULL,                 # Group ID
  PRIMARY KEY     (id)                                  # Make the id the primary key
);
```

Database
information_schema
meetup
mysql
performance_schema
sys

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#   Time      Action                                                                          Message             Duration / Fetch
17  12:23:24  SHOW TABLES                                                                     1 row(s) returned   0.000 sec / 0.000 sec
18  12:23:59  USE meetup                                                                      0 row(s) affected   0.000 sec
19  12:23:59  DESCRIBE login                                                                  3 row(s) returned   0.016 sec / 0.000 sec
20  12:28:11  USE meetup                                                                      0 row(s) affected   0.000 sec
21  12:28:11  CREATE TABLE users (  id   INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the recor...   0 row(s) affected   0.031 sec
22  12:28:11  SHOW DATABASES                                                                  5 row(s) returned   0.000 sec / 0.000 sec
```



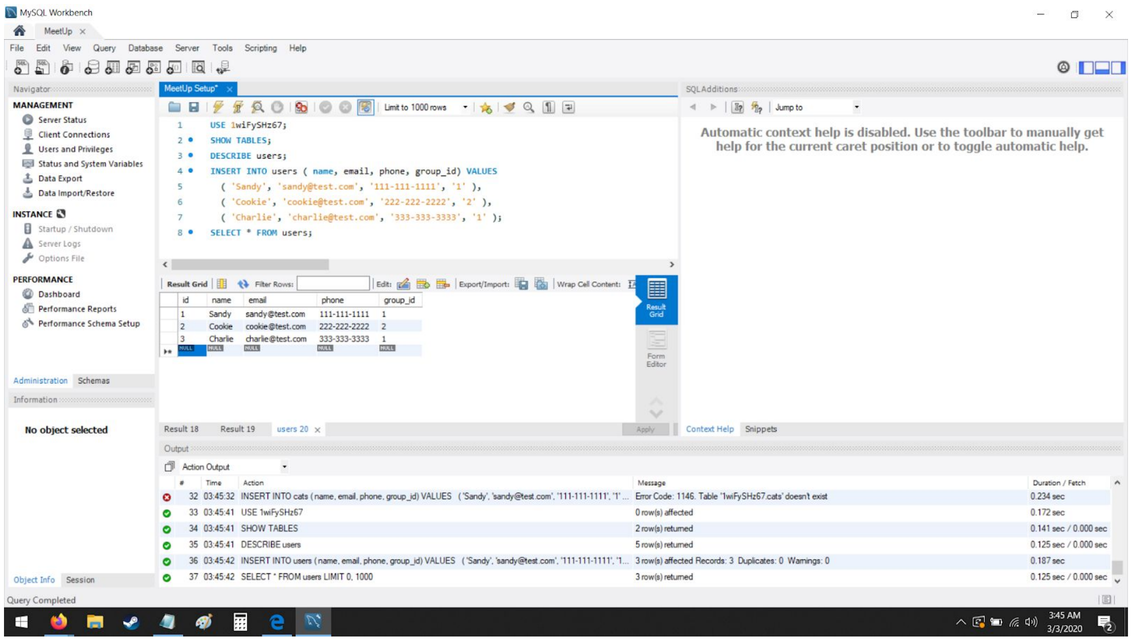
```
USE meetup;
DESCRIBE users;
```

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
name	varchar(150)	NO		NULL	
email	varchar(150)	NO		NULL	
phone	varchar(150)	NO		NULL	
group_id	int unsigned	NO		NULL	

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#   Time      Action                                                                                        Message              Duration / Fetch
19  12:23:59  DESCRIBE login                                                                                3 row(s) returned    0.016 sec / 0.000 sec
20  12:28:11  USE meetup                                                                                    0 row(s) affected    0.000 sec
21  12:28:11  CREATE TABLE users (  id        INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the recor...  0 row(s) affected    0.031 sec
22  12:28:11  SHOW DATABASES                                                                                5 row(s) returned    0.000 sec / 0.000 sec
23  12:28:44  USE meetup                                                                                    0 row(s) affected    0.000 sec
24  12:28:44  DESCRIBE users                                                                                5 row(s) returned    0.000 sec / 0.000 sec
```



```
USE 1wiFySHz67;
SHOW TABLES;
DESCRIBE users;
INSERT INTO users ( name, email, phone, group_id) VALUES
  ( 'Sandy', 'sandy@test.com', '111-111-1111', '1' ),
  ( 'Cookie', 'cookie@test.com', '222-222-2222', '2' ),
  ( 'Charlie', 'charlie@test.com', '333-333-3333', '1' );
SELECT * FROM users;
```

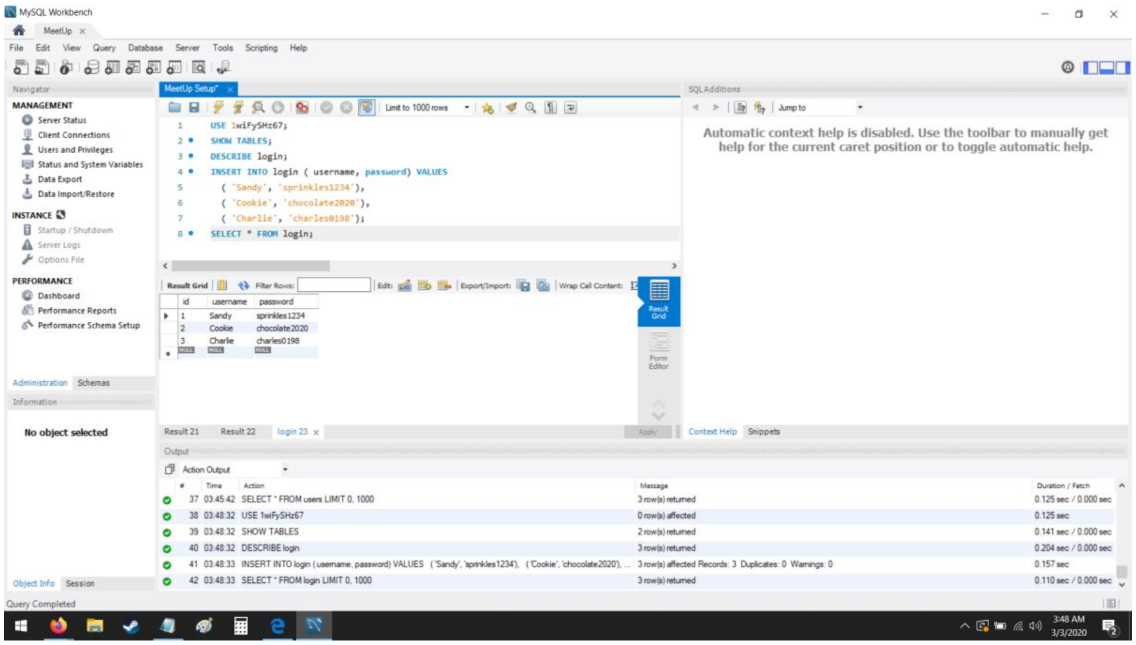
id	name	email	phone	group_id
1	Sandy	sandy@test.com	111-111-1111	1
2	Cookie	cookie@test.com	222-222-2222	2
3	Charlie	charlie@test.com	333-333-3333	1
NULL	NULL	NULL	NULL	NULL

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#   Time      Action                                                                                          Message                                                    Duration / Fetch
32  03:45:32  INSERT INTO cats (name, email, phone, group_id) VALUES  ('Sandy', 'sandy@test.com', '111-111-1111', '1' ...  Error Code: 1146. Table '1wiFySHz67.cats' doesn't exist      0.234 sec
33  03:45:41  USE 1wiFySHz67                                                                                  0 row(s) affected                                          0.172 sec
34  03:45:41  SHOW TABLES                                                                                     2 row(s) returned                                          0.141 sec / 0.000 sec
35  03:45:41  DESCRIBE users                                                                                  5 row(s) returned                                          0.125 sec / 0.000 sec
36  03:45:42  INSERT INTO users (name, email, phone, group_id) VALUES  ('Sandy', 'sandy@test.com', '111-111-1111', '1...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0    0.187 sec
37  03:45:42  SELECT * FROM users LIMIT 0, 1000                                                               3 row(s) returned                                          0.125 sec / 0.000 sec
```

Query Completed



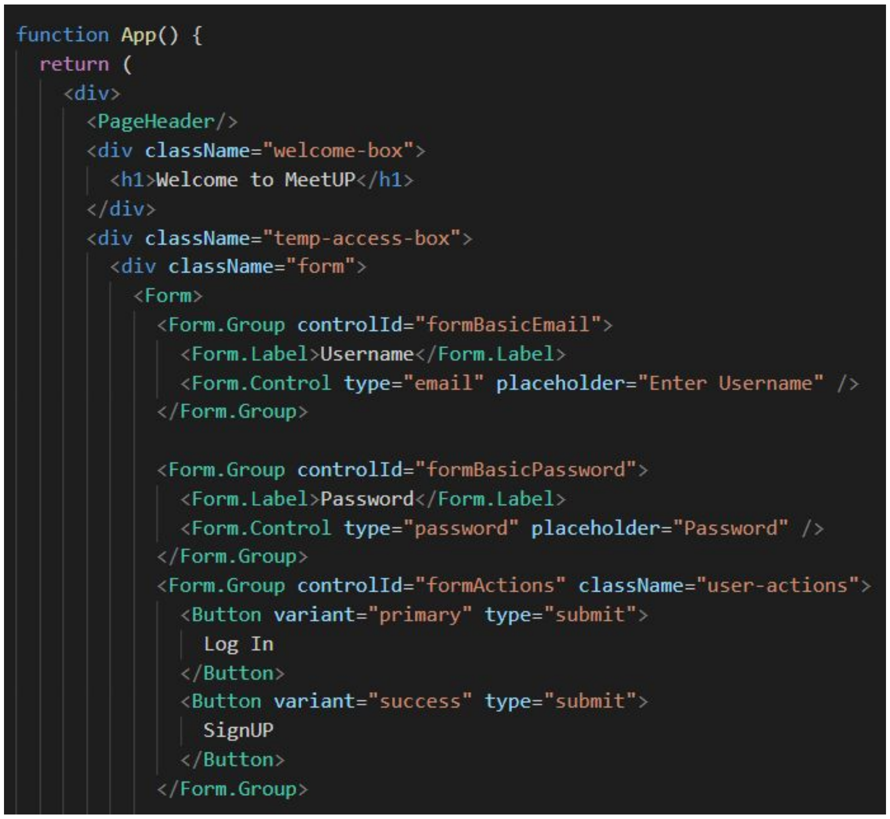
```sql
USE 1wiFySHz67;
SHOW TABLES;
DESCRIBE login;
INSERT INTO login ( username, password) VALUES
  ( 'Sandy', 'sprinkles1234'),
  ( 'Cookie', 'chocolate2020'),
  ( 'Charlie', 'charles0198');
SELECT * FROM login;
```

id	username	password
1	Sandy	sprinkles1234
2	Cookie	chocolate2020
3	Charlie	charles0198
NULL	NULL	NULL

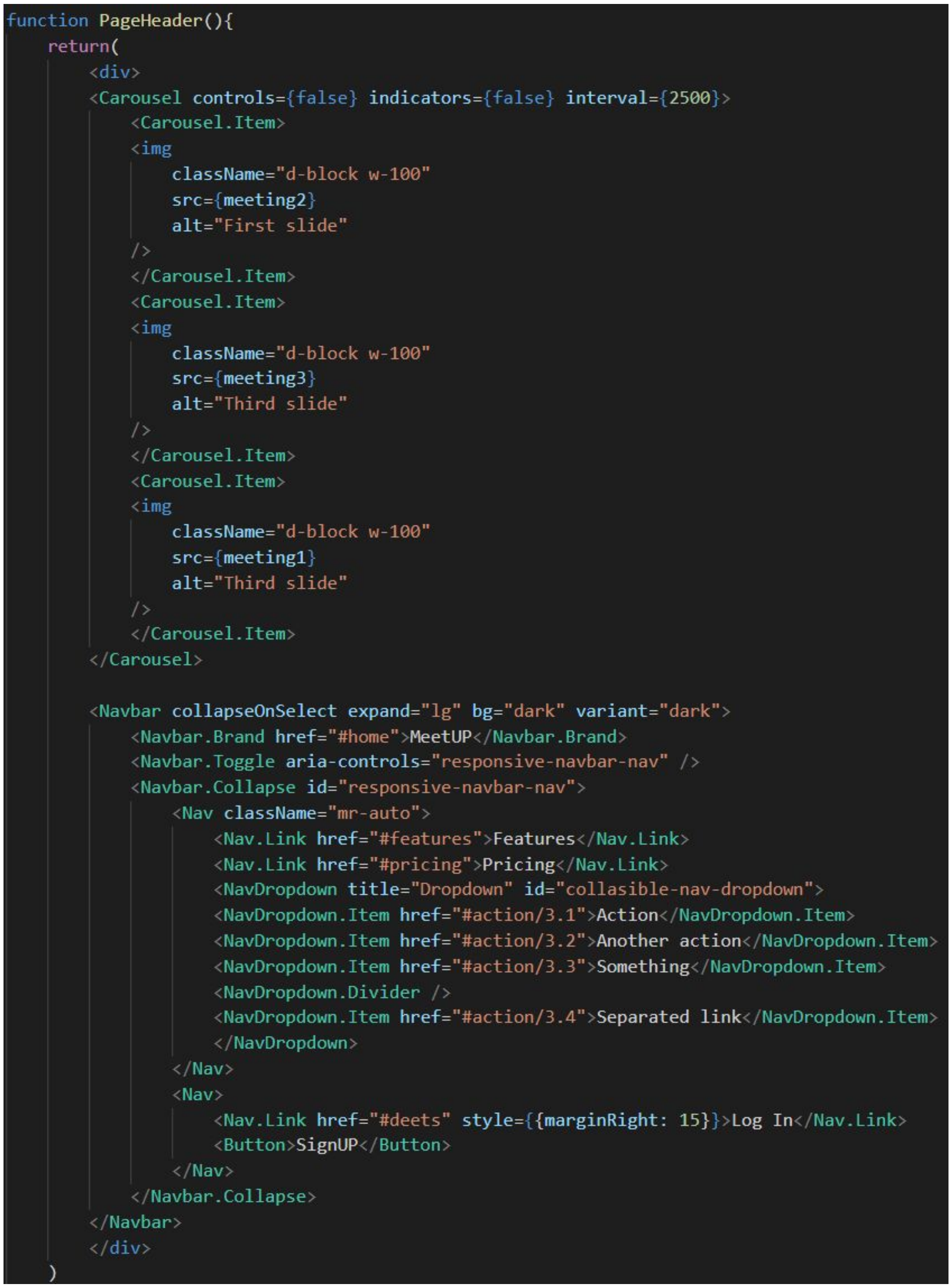
```
 #  Time      Action                                                                                              Message                                                Duration / Fetch
37  03:45:42  SELECT * FROM users LIMIT 0, 1000                                                                   3 row(s) returned                                      0.125 sec / 0.000 sec
38  03:48:32  USE 1wiFySHz67                                                                                      0 row(s) affected                                      0.125 sec
39  03:48:32  SHOW TABLES                                                                                         2 row(s) returned                                      0.141 sec / 0.000 sec
40  03:48:32  DESCRIBE login                                                                                      3 row(s) returned                                      0.204 sec / 0.000 sec
41  03:48:33  INSERT INTO login (username, password) VALUES  ('Sandy', 'sprinkles1234'),  ('Cookie', 'chocolate2020'), ...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.157 sec
42  03:48:33  SELECT * FROM login LIMIT 0, 1000                                                                   3 row(s) returned                                      0.110 sec / 0.000 sec
```

The database is used to store usernames, emails, phone numbers and passwords for use in the login and register use cases. In order to test the database, we used a MySQL workbench server. This will not be the final hosting service, but it was useful for initial implementation.

Frontend and Logic:



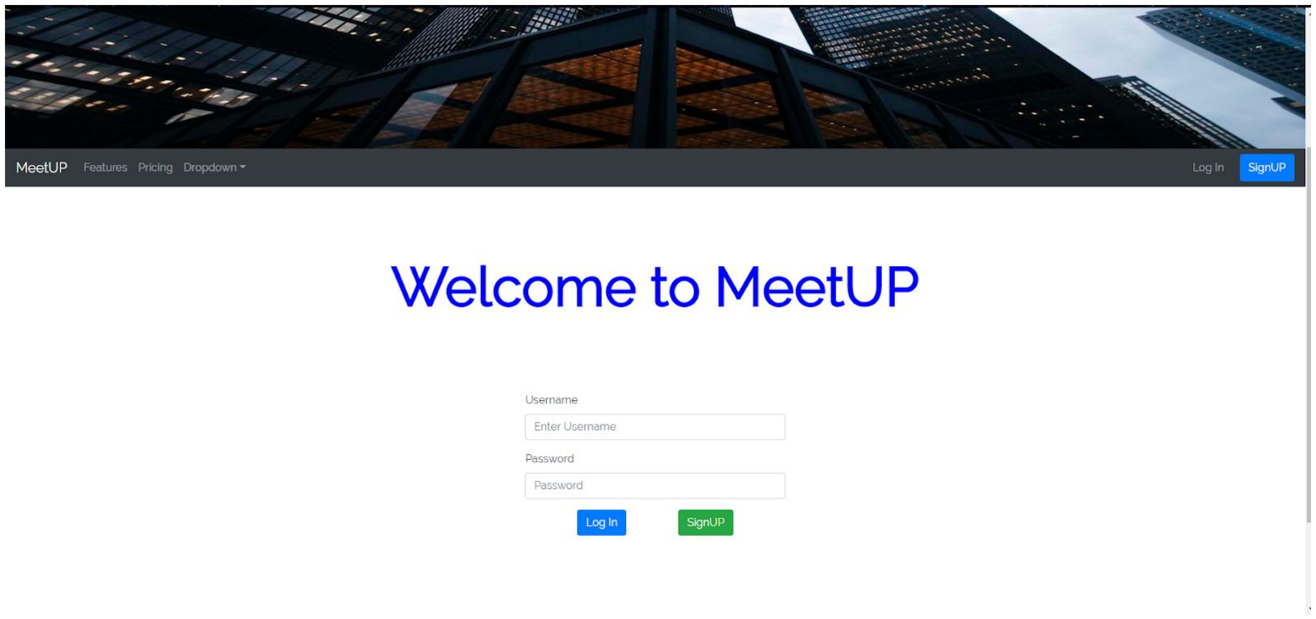
```jsx
function App() {
  return (
    <div>
      <PageHeader/>
      <div className="welcome-box">
        <h1>Welcome to MeetUP</h1>
      </div>
      <div className="temp-access-box">
        <div className="form">
          <Form>
            <Form.Group controlId="formBasicEmail">
              <Form.Label>Username</Form.Label>
              <Form.Control type="email" placeholder="Enter Username" />
            </Form.Group>

            <Form.Group controlId="formBasicPassword">
              <Form.Label>Password</Form.Label>
              <Form.Control type="password" placeholder="Password" />
            </Form.Group>
            <Form.Group controlId="formActions" className="user-actions">
              <Button variant="primary" type="submit">
                Log In
              </Button>
              <Button variant="success" type="submit">
                SignUP
              </Button>
            </Form.Group>
```



```
function PageHeader(){
    return(
        <div>
        <Carousel controls={false} indicators={false} interval={2500}>
            <Carousel.Item>
            <img
                className="d-block w-100"
                src={meeting2}
                alt="First slide"
            />
            </Carousel.Item>
            <Carousel.Item>
            <img
                className="d-block w-100"
                src={meeting3}
                alt="Third slide"
            />
            </Carousel.Item>
            <Carousel.Item>
            <img
                className="d-block w-100"
                src={meeting1}
                alt="Third slide"
            />
            </Carousel.Item>
        </Carousel>

        <Navbar collapseOnSelect expand="lg" bg="dark" variant="dark">
            <Navbar.Brand href="#home">MeetUP</Navbar.Brand>
            <Navbar.Toggle aria-controls="responsive-navbar-nav" />
            <Navbar.Collapse id="responsive-navbar-nav">
                <Nav className="mr-auto">
                    <Nav.Link href="#features">Features</Nav.Link>
                    <Nav.Link href="#pricing">Pricing</Nav.Link>
                    <NavDropdown title="Dropdown" id="collasible-nav-dropdown">
                    <NavDropdown.Item href="#action/3.1">Action</NavDropdown.Item>
                    <NavDropdown.Item href="#action/3.2">Another action</NavDropdown.Item>
                    <NavDropdown.Item href="#action/3.3">Something</NavDropdown.Item>
                    <NavDropdown.Divider />
                    <NavDropdown.Item href="#action/3.4">Separated link</NavDropdown.Item>
                    </NavDropdown>
                </Nav>
                <Nav>
                    <Nav.Link href="#deets" style={{marginRight: 15}}>Log In</Nav.Link>
                    <Button>SignUP</Button>
                </Nav>
            </Navbar.Collapse>
        </Navbar>
        </div>
    )
```

```
import javax.validation.Valid;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.Collection;
import java.util.Optional;



@RestController
@RequestMapping("/api")
class UserController{

    private UserRepository userRepository;

    public UserController(UserRepository userRepository){
        this.userRepository = userRepository;
    }

    @GetMapping(path="/login/{username}")
    ResponseEntity<?> getUser(@PathVariable String username){
        Optional<User> user = userRepository.findByUsername(username);
        return user.map(response -> ResponseEntity.ok().body(response)).orElse(new ResponseEntity<>(HttpStatus.NOT_FOUND));
    }
}
```

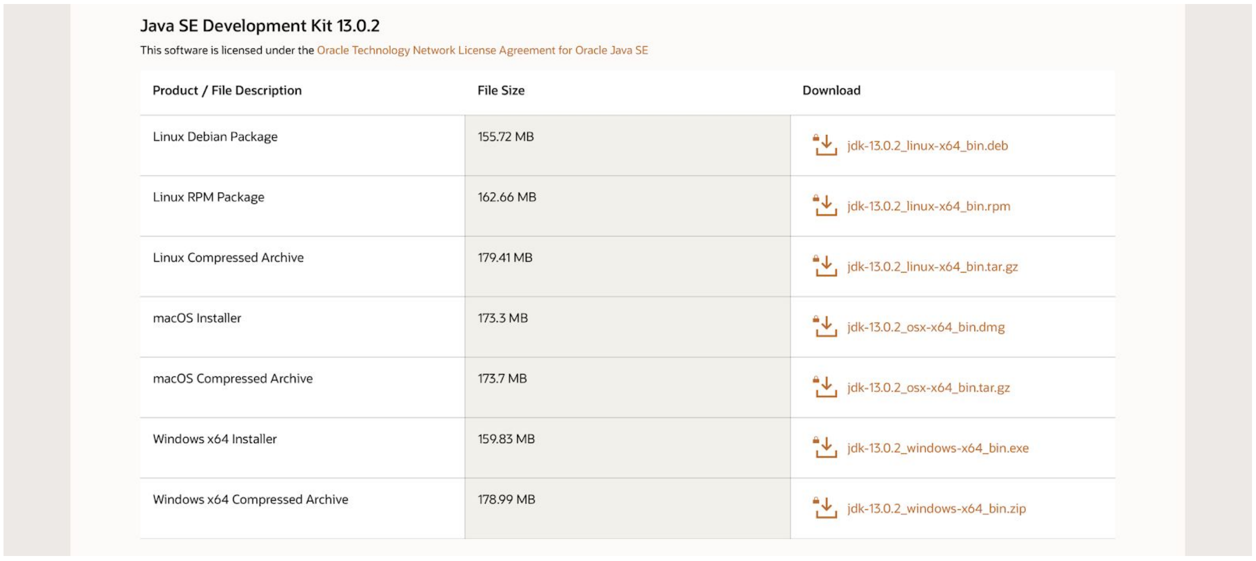
The two use cases that we developed a front-end and database for where the register and login use cases.



Testing

Testing Java in Visual Studio Code can be enabled by the Java Test Runner extension. The extension supports JUnit 4, JUnit 5, and TestNG. It can run/debug test cases with customizable configurations and show test logs and reports.

First, Java Development Kit should be installed.



Java SE Development Kit 13.0.2

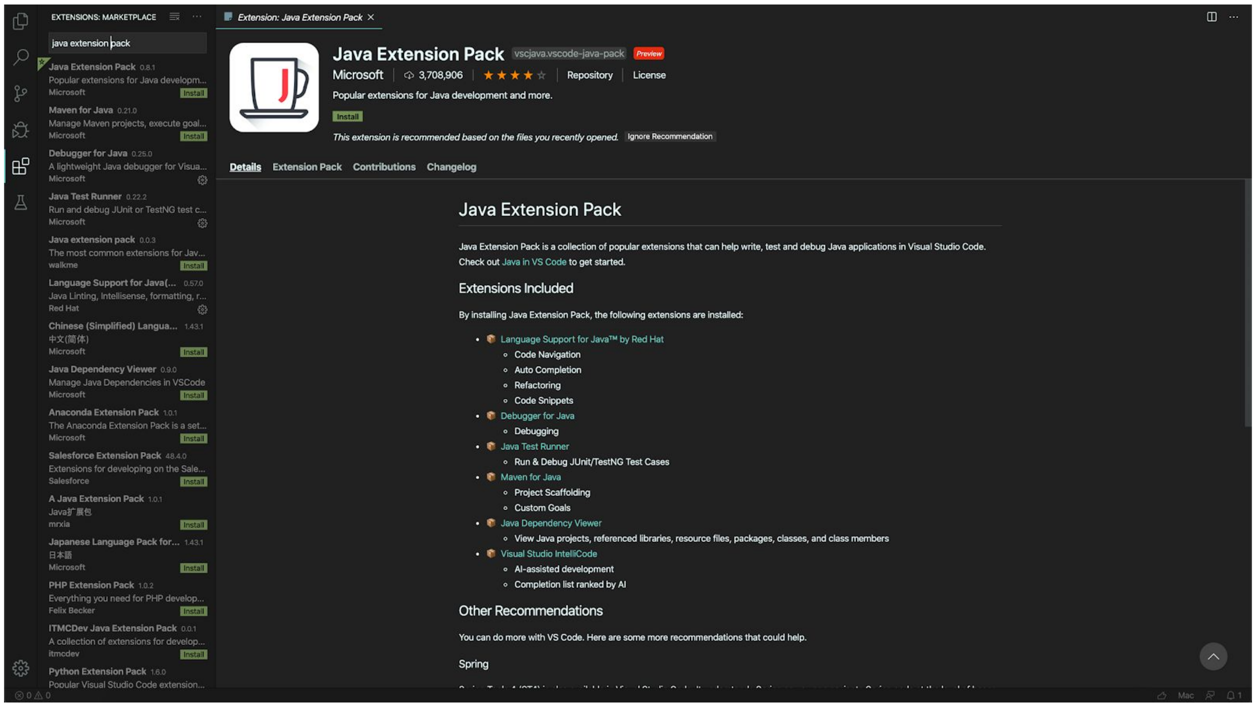
This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE

Product / File Description	File Size	Download
Linux Debian Package	155.72 MB	jdk-13.0.2_linux-x64_bin.deb
Linux RPM Package	162.66 MB	jdk-13.0.2_linux-x64_bin.rpm
Linux Compressed Archive	179.41 MB	jdk-13.0.2_linux-x64_bin.tar.gz
macOS Installer	173.3 MB	jdk-13.0.2_osx-x64_bin.dmg
macOS Compressed Archive	173.7 MB	jdk-13.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	159.83 MB	jdk-13.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	178.99 MB	jdk-13.0.2_windows-x64_bin.zip

Assuming Visual Studio Code is ready, the following extension package can be installed, which includes all of these necessary extensions:

1) Language Support for Java by Red Hat

2) Debugger for Java
3) Java Test Runner



EXTENSIONS: MARKETPLACE

java extension pack

Java Extension Pack 0.8.1
Popular extensions for Java developm...
Microsoft — Install

Maven for Java 0.21.0
Manage Maven projects, execute goal...
Microsoft — Install

Debugger for Java 0.25.0
A lightweight Java debugger for Visua...
Microsoft

Java Test Runner 0.22.2
Run and debug JUnit or TestNG test c...
Microsoft

Java extension pack 0.0.3
The most common extensions for Jav...
walkme — Install

Language Support for Java(... 0.57.0
Java Linting, Intellisense, formatting, r...
Red Hat

Chinese (Simplified) Langua... 1.43.1
中文(简体)
Microsoft — Install

Java Dependency Viewer 0.9.0
Manage Java Dependencies in VSCode
Microsoft — Install

Anaconda Extension Pack 1.0.1
The Anaconda Extension Pack is a set...
Microsoft — Install

Salesforce Extension Pack 48.4.0
Extensions for developing on the Sale...
Salesforce — Install

A Java Extension Pack 1.0.1
Java扩展包
mrxia — Install

Japanese Language Pack for... 1.43.1
日本語
Microsoft — Install

PHP Extension Pack 1.0.2
Everything you need for PHP develop...
Felix Becker — Install

ITMCDev Java Extension Pack 0.0.1
A collection of extensions for develop...
itmcdev — Install

Python Extension Pack 1.6.0
Popular Visual Studio Code extension...

Extension: Java Extension Pack

Java Extension Pack vscjava.vscode-java-pack Preview
Microsoft | 3,708,906 | ★★★★☆ | Repository | License
Popular extensions for Java development and more.
Install
This extension is recommended based on the files you recently opened. Ignore Recommendation

Details Extension Pack Contributions Changelog

Java Extension Pack

Java Extension Pack is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out Java in VS Code to get started.

Extensions Included

By installing Java Extension Pack, the following extensions are installed:

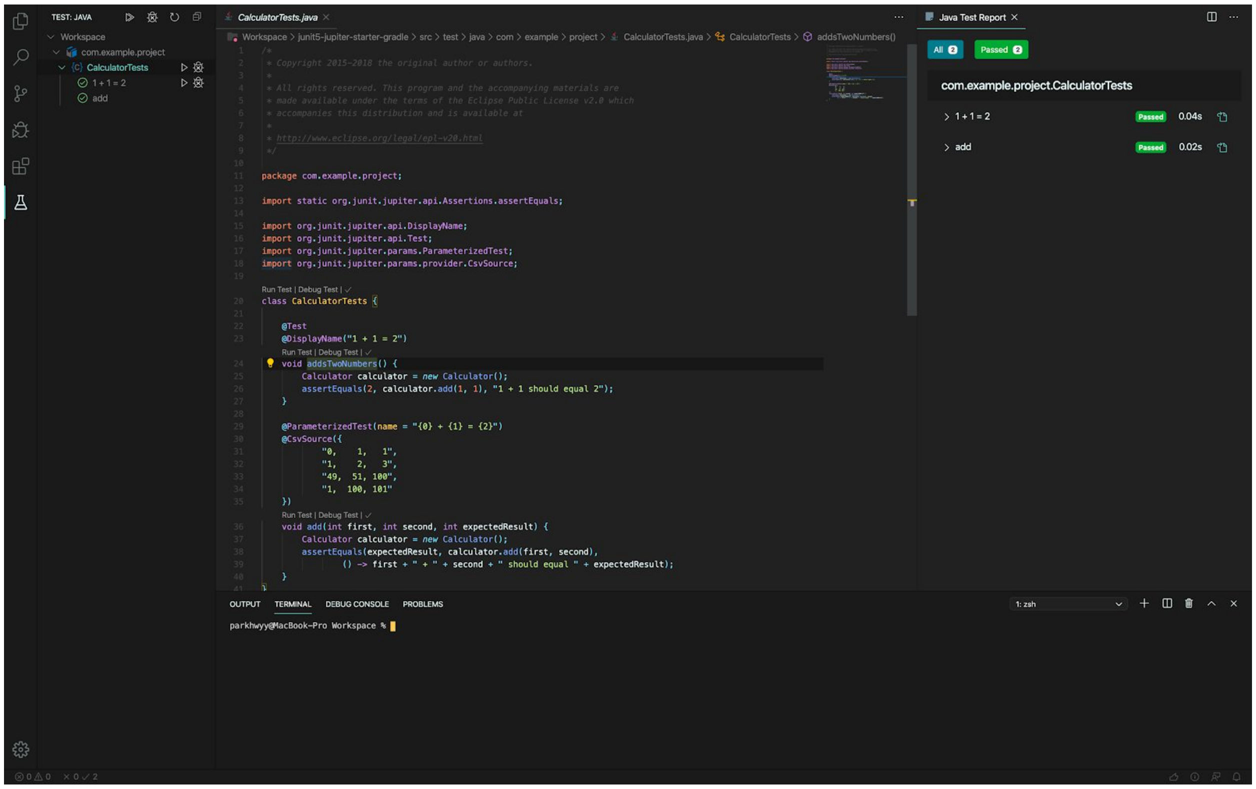
- Language Support for Java™ by Red Hat
  - Code Navigation
  - Auto Completion
  - Refactoring
  - Code Snippets
- Debugger for Java
  - Debugging
- Java Test Runner
  - Run & Debug JUnit/TestNG Test Cases
- Maven for Java
  - Project Scaffolding
  - Custom Goals
- Java Dependency Viewer
  - View Java projects, referenced libraries, resource files, packages, classes, and class members
- Visual Studio IntelliCode
  - AI-assisted development
  - Completion list ranked by AI

Other Recommendations

You can do more with VS Code. Here are some more recommendations that could help.

Spring

The following screenshot captures a sample test using JUnit in Visual Studio Code.



TEST: JAVA
Workspace
com.example.project
CalculatorTests
1 + 1 = 2
add

CalculatorTests.java
Workspace > junit5-jupiter-starter-gradle > src > test > java > com > example > project > CalculatorTests.java > CalculatorTests > addsTwoNumbers()

```java
/*
 * Copyright 2015-2018 the original author or authors.
 *
 * All rights reserved. This program and the accompanying materials are
 * made available under the terms of the Eclipse Public License v2.0 which
 * accompanies this distribution and is available at
 *
 * http://www.eclipse.org/legal/epl-v20.html
 */

package com.example.project;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;

Run Test | Debug Test | ✓
class CalculatorTests {

    @Test
    @DisplayName("1 + 1 = 2")
    Run Test | Debug Test | ✓
    void addsTwoNumbers() {
        Calculator calculator = new Calculator();
        assertEquals(2, calculator.add(1, 1), "1 + 1 should equal 2");
    }

    @ParameterizedTest(name = "{0} + {1} = {2}")
    @CsvSource({
            "0,    1,   1",
            "1,    2,   3",
            "49,  51, 100",
            "1,  100, 101"
    })
    Run Test | Debug Test | ✓
    void add(int first, int second, int expectedResult) {
        Calculator calculator = new Calculator();
        assertEquals(expectedResult, calculator.add(first, second),
                () -> first + " + " + second + " should equal " + expectedResult);
    }
}
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
parkhwyy@MacBook-Pro Workspace %
```

Java Test Report

All 2 Passed 2

com.example.project.CalculatorTests

```
> 1 + 1 = 2    Passed  0.04s
> add          Passed  0.02s
```

Use Case 1: Register Account

Function: register (String username, String password)

Input partition: unique | existing | empty (for username)

valid | invalid | empty (for password)

Test specifications:	Test cases:
register (unique username, valid pw)	register (“user00”, “User001*”)
register (existing username, valid pw)	register (“admin”, “User001*”)
register (unique username, invalid pw)	register (“user00”, “user00”)
register (existing username, invalid pw)	register (“admin”, “user00”)
register (unique username, empty pw)	register (“user00”, “”)
register (existing username, empty pw)	register (“admin”, “”)
register (empty username, valid pw)	register (“”, “User001*”)
register (empty username, invalid pw)	register (“”, “user00”)
register (empty username, empty pw)	register (“”, “”)

Use Case 2: Login to Account

Function: login (String username, String password)

Input partition: correct | incorrect | empty (for both username and password)

Test specifications:

login (correct username, correct pw)

login (incorrect username, correct pw)

login (correct username, incorrect pw)

login (incorrect username, incorrect pw)

login (correct username, empty pw)

login (incorrect username, empty pw)

login (empty username, correct pw)

login (empty username, incorrect pw)

login (empty username, empty pw)

Test cases:

login (“admin”, “Admin001*”)

login (“admit”, “Admin001*”)

login (“admin”, “Admin00”)

login (“admit”, “Admin00”)

login (“admin”, “”)

login (“admit”, “”)

login (“”, “Admin001*”)

login (“”, “Admin00”)	login (“”, “”)

Appendix

Youtube link
https://www.youtube.com/watch?v=JsBMbk5mbAU&feature=youtu.be

Discord link
https://discordapp.com/channels/670083895097688087/670083895097688099

Github Repository Link
https://github.com/ConnorLacy/CSC-SWE-CEDAH/tree/master/src