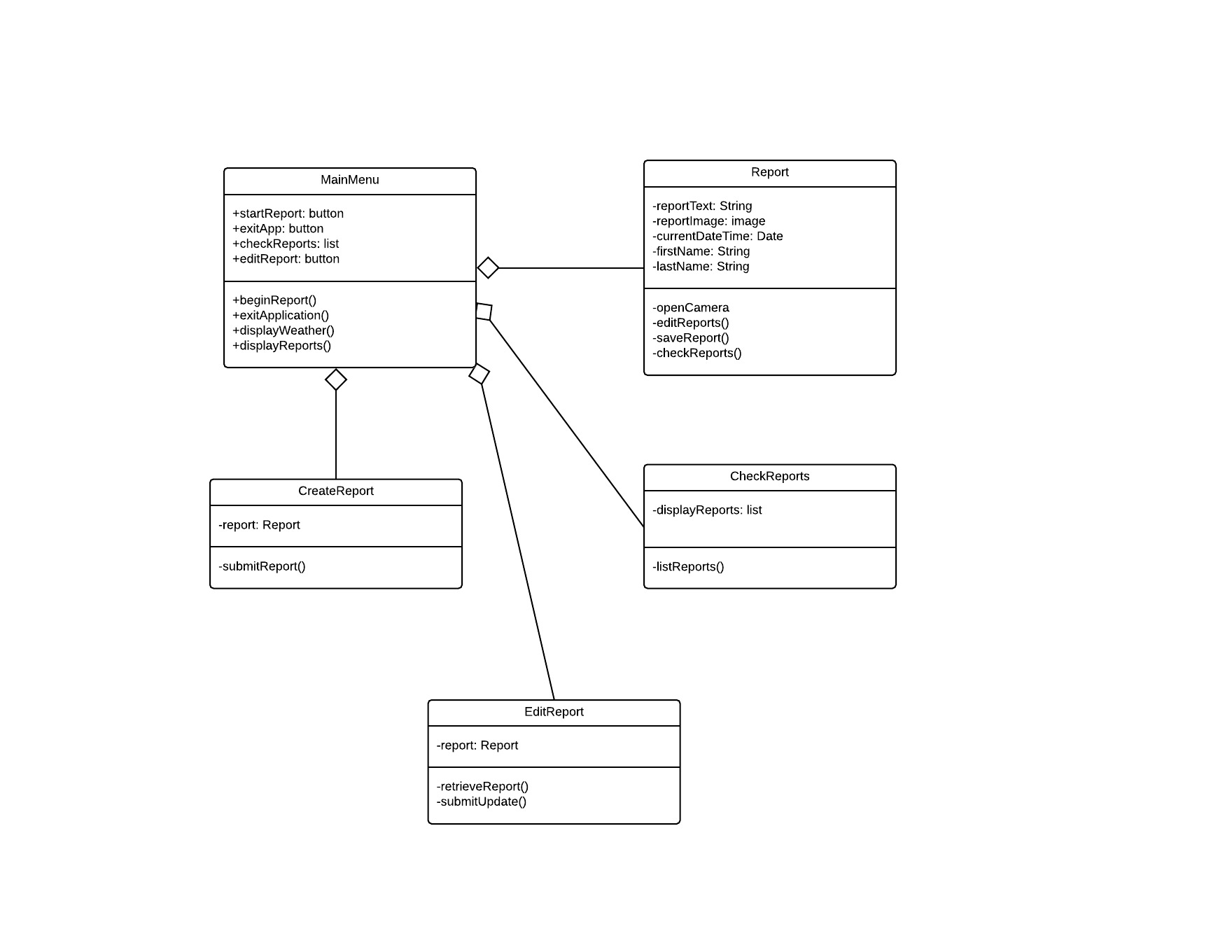
Quick Reports App: Design

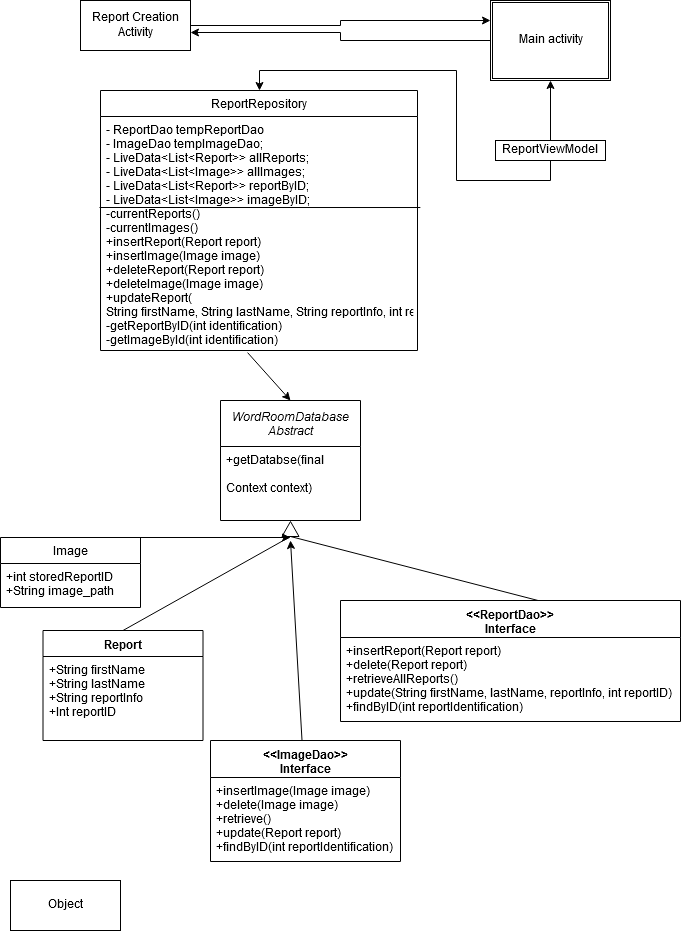
Jesse Willams, James Jasinski, Matthew Coleman

UML diagram 1

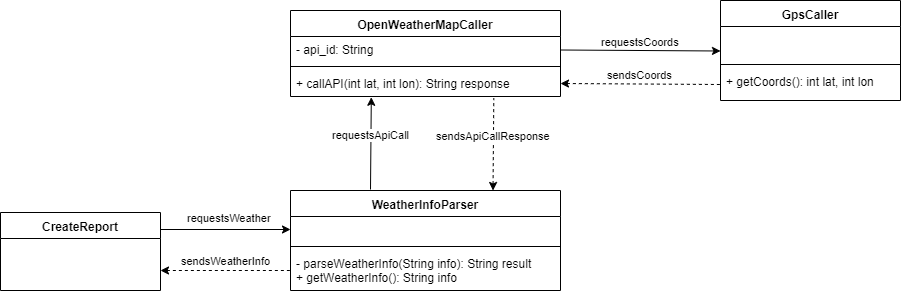


This handles the part of the application that shows the main menu to the user and allows the user to create reports, edit a report and show a list of reports already submitted, as well as allowing users to update an existing report. Also handles closing the app. This will use the database part of the system which is shown below.

Database Module:

The following UML diagram represents the backend database being used for the Quick Reports application. There are six major Classes to the UML diagram. Image, Report, ReportDao, and ImageDao function as the creation of tables and queries within the database. They essentially make up the filling of the database and fill out the list of entities inside the database. The database class is abstracted because it is used more as a constructor for the database while the four other classes mentioned above are the real inner workings of the database module. Next there is the biggest part of the database, the repository. The repository is used as a sort of encapsulation for the database, acting as the go between for the main activity and the backend database. While the repository is not truly needed it helps protect the database and ensure the inner workings of the database are shielded from anything the user may try and do. In the end the repository houses all the functions that are available to the user and runs them against the database when it is prompted to.

Weather API Access:



On creation of a report, the OpenWeatherMap API is called to retrieve the current weather at the user’s location. The API’s response is used to generate a list of weather information which is saved with the report.

**Development Plan**:

Responsibility list: James will be responsible for the development of the module that will allow our application to access an outside weather API. Jesse will be responsible for the module that will handle creating reports as well as the UI connected to said module. Matthew will be responsible for the database that will be used to store both the reports and the image paths we will be using to store the images. We will be primarily using text messages to communicate and set up meeting days.

Schedule: The UML document will be due on the 21st of November so we can finalize it and continue our work on the project. From that point on we will be meeting every Tuesday and Thursday so that a set schedule of updates is established. We plan to have the database done and tested by the 24th. From there on all the other products are due by the 3rd of December will updates every Tuesday and Thursday as described above.

Documentation plan: An instruction document will be created to help guide the user through the use of the Quick Reports application and documentation will happen concurrently with development. Alongside the readme report we will include JavaDoc comments within the code for each piece of code that needs to be explained, such as specific functionality that may not be clear on first inspection.