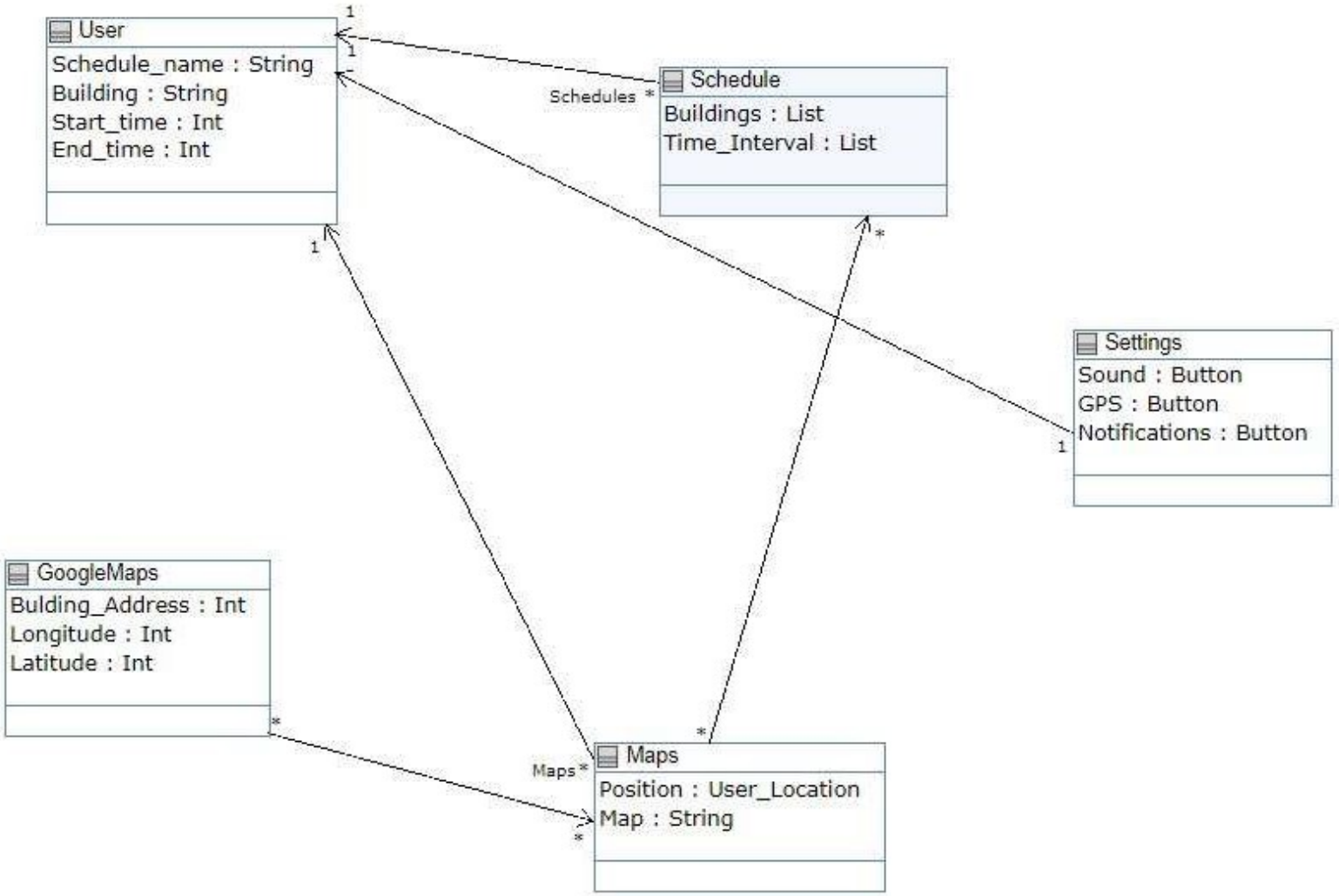
Domain Model

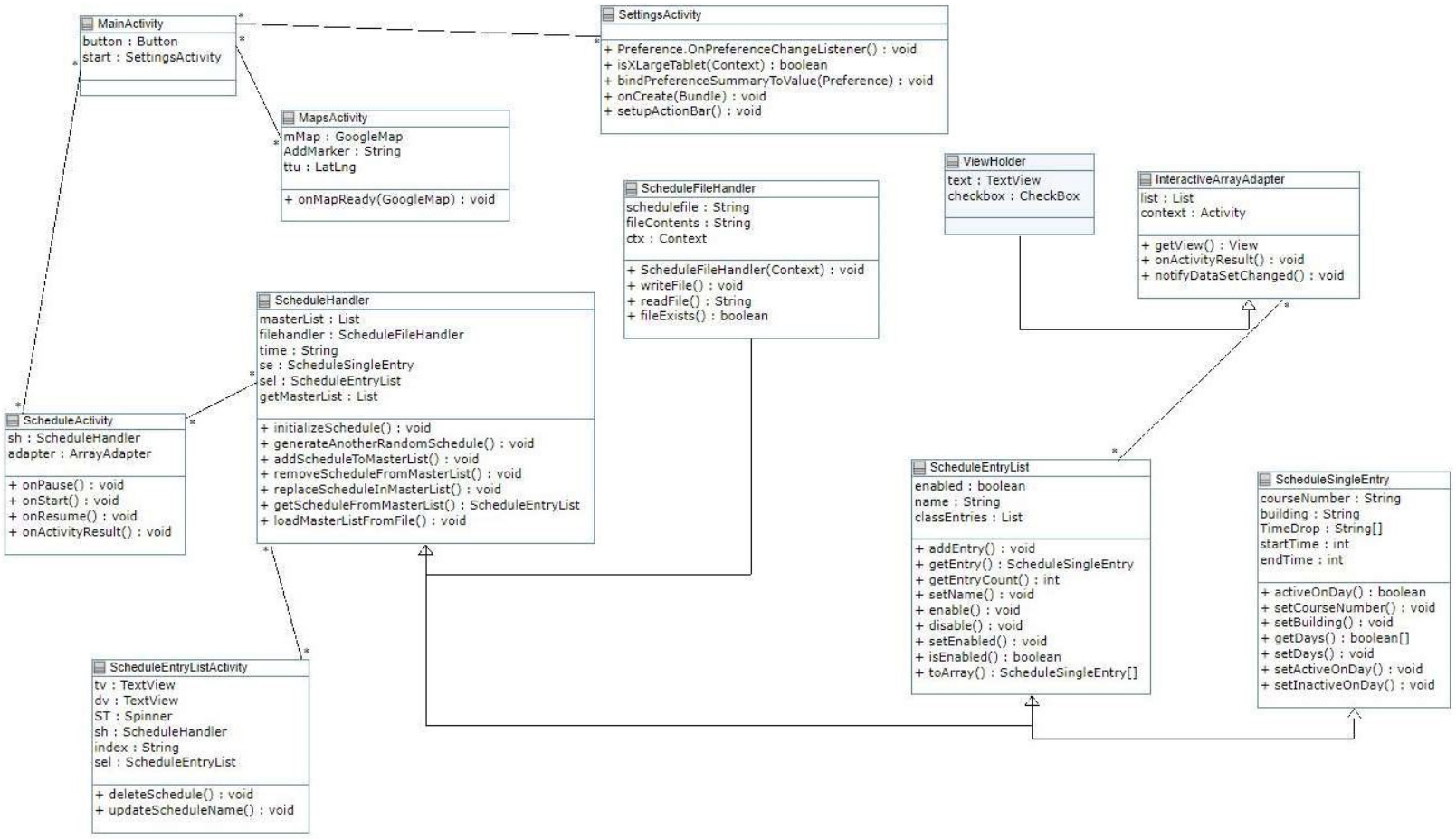
The Domain Model is a high level diagram of the major entities of our RaiderNAV application. This model shows the attributes, connections, and the multiplicity of the entities to allow there to be a better understanding of the application's overall operations. Each entity contains a list of properties that they will provide the application to attain the goal of the application. This high level domain model is a basis of RaiderNAV and provides the company a foundation of each concept of the application.



http://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=180309781312

Detailed Design Model Document

The Detailed Design Model Document is designed to closely mirror exactly what will be implemented in the code of the application. Each class is represented as it's own entity, containing all of its important variables and functions that it calls. The entities are then connected by lines that show how information is connected through classes and which classes are Super classes or sub classes, bringing to the model a sense of dependency between the classes. This model is highly important in depicting each class that is necessary for the application to run, while expressing each and every attribute of the classes. Over all, this model will be able to translate to code and is a critical process to developing RaiderNAV.



http://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=18030850486

Umple Domain Model Code

```
class User
{String Schedule_name;
 String Building;
 Int Start_time;
 Int End_time;
 }

class Schedule
{

 List Buildings;
 List Time_Interval;
*Schedules -> 1 User;
}

class Settings
{
 Button Sound;
 Button GPS;
 Button Notifications;

 1 -> 1 User;
}

class Maps
{
 User_Location Position;
 Map;
 *Maps  ->1 User;
 * ->* Schedule;
}

class GoogleMaps
{
 Int Bulding_Address;
 Int Longitude;
 Int Latitude;
 *-> * Maps;
}
```

Umple Detailed Design Model Document

```
class MainActivity{
   Button button;
  SettingsActivity start;

      * -- * ScheduleActivity;
  * -- * MapsActivity;
  * -- * SettingsActivity;
}



class MapsActivity{
GoogleMap mMap;
  AddMarker;
  public void moveCamera(CameraUpdateFactory.newLatLng(ttu)){
  }
  LatLng ttu = new LatLng(33.584468, -101.874658);
public void onMapReady(GoogleMap googleMap) {
  }
}
class ScheduleActivity{
   ScheduleHandler sh;
   ArrayAdapter<ScheduleEntryList> adapter;
   * -- * ScheduleHandler;
public void onPause(){
  }
  public void onStart(){
  }
  public void onResume(){
  }
  public void onActivityResult(){
  }
}
class ScheduleEntryList {
     isA ScheduleSingleEntry;
boolean enabled;
   String name;
   List<ScheduleSingleEntry> classEntries;
```

```
   public void addEntry(){
  }
  public ScheduleSingleEntry getEntry(){
  }
  int getEntryCount(){
  }
  String getName(){
  }
  void setName(){
  }
  void enable(){
  }
  void disable(){
  }
  void setEnabled(){
  }
  boolean isEnabled(){
  }
  ScheduleSingleEntry[] toArray(){
  }
}
class ScheduleEntryListActivity{
    TextView tv;
    TextView dv;
    Spinner ST;
    ScheduleHandler sh;
    index = extras.getInt("index");
    ScheduleEntryList sel = sh.getMasterList().get(index);
      * -- * ScheduleHandler;
void deleteSchedule(){
  }
  void updateScheduleName(){
  }

}

class ScheduleFileHandler {
     isA ScheduleHandler;
String schedulefile = "schedulefile";
```

```
   String fileContents = "Initial text";
   Context ctx;

   ScheduleFileHandler(Context c) {
       ctx = c;
  }
  void writeFile(){
       FileOutputStream outputStream;
  }
  String readFile(){
  }
  boolean fileExists(){
  }

}

class ScheduleHandler {
     isA ScheduleEntryList;
List<ScheduleEntryList> masterList;
   ScheduleFileHandler filehandler;
  void initializeSchedule(){
  }
        String time;
        ScheduleSingleEntry se = new ScheduleSingleEntry("CS3365", "LIVERMORE", time, 1050);
       ScheduleEntryList sel = new ScheduleEntryList("tuethur-spring18");
       sel.addEntry(se);

  void generateAnotherRandomSchedule(){
  }
  void addScheduleToMasterList(){
  }
  void removeScheduleFromMasterList(){
  }
  void replaceScheduleInMasterList(){
  }
  ScheduleEntryList getScheduleFromMasterList(){
  }
  void loadMasterListFromFile(){
  }
```

```
  List<ScheduleEntryList> getMasterList;

}

class ScheduleSingleEntry {
    String courseNumber;
    String building;
    String [] TimeDrop;
    int startTime;
    int endTime;
  boolean activeOnDay(){
  }
  void setCourseNumber(){
  }
  String getCourseNumber(){
  }
  void setBuilding(){
  }
  String [] getTimeDrop(){
  }
  int getEndTime(){
  }
  boolean[] getDays(){
  }
  void setDays(){
  }
  void setActiveOnDay(){
  }
  void setInactiveOnDay(){
  }

}

class InteractiveArrayAdapter {

    List<ScheduleEntryList> list;
    Activity context;
    * -- * ScheduleEntryList;
class ViewHolder {
```

```
        TextView text;
        CheckBox checkbox;
    }
  public View getView(){
  }
  void onActivityResult(){
  }
  void notifyDataSetChanged(){
  }
}


class SettingsActivity{
  Preference.OnPreferenceChangeListener(){}
  boolean isXLargeTablet(Context context) {
  }
  bindPreferenceSummaryToValue(Preference preference) {
  }
  onCreate(Bundle savedInstanceState) {
  }
  setupActionBar() {
  }
}
```