**Invisino**

**Part 1: Requirements Documentation**

**CS 495**

**Submitted by:**

**Maggie Gembala**

**Nicole Tollman**

**Jessie Melton**

**Justin Estep**

**The University of Alabama**

**Tuscaloosa, Alabama 35487**

**October 3, 2017**

**Update History:**

* Creation, Set up outline – September 19, 2017 Maggie Gembala
* Added activity diagrams – September 24, 2017 Maggie Gembala
* Added use case diagrams – September 25, 2017 Maggie Gembala

**Table of Contents:**

Section Page Number

1. [Update History](#Update_History)
2. [Table of Contents](#Table_of_Contents)
3. [Introduction](#Introduction)
   1. [Motivation](#Motivation)
   2. [Scope](#Scope)
   3. [Goals](#Goals)
   4. [Relevant Definitions](#Relevant_Definitions)
4. [Project Description](#Project_Description)
   1. [Primary Features](#Primary_Features)
   2. [Secondary Features](#Secondary_Features)
5. [Summary of Requirements](#Summary_of_Requirements)
   1. [Functional Requirements](#Functional_Requirements)
   2. [Nonfunctional Requirements](#Nonfunctional_Requirements)
6. [Diagrams](#Diagrams)
   1. [Use Case Diagrams](#Use_Case_Diagrams)
   2. [Activity Diagrams](#Activity_Diagrams)
   3. [Class Diagrams](#Class_Diagrams)
      1. [High Level](#High_Level_Class_Diagram)
      2. [Detailed](#Detailed_Class_Diagram)
   4. [Sequence Diagrams](#Sequence_Diagrams)

**I Introduction:**

1. **Motivation**
2. **Scope**
3. **Goals**
4. **Relevant Definitions**

**II Project Description:**

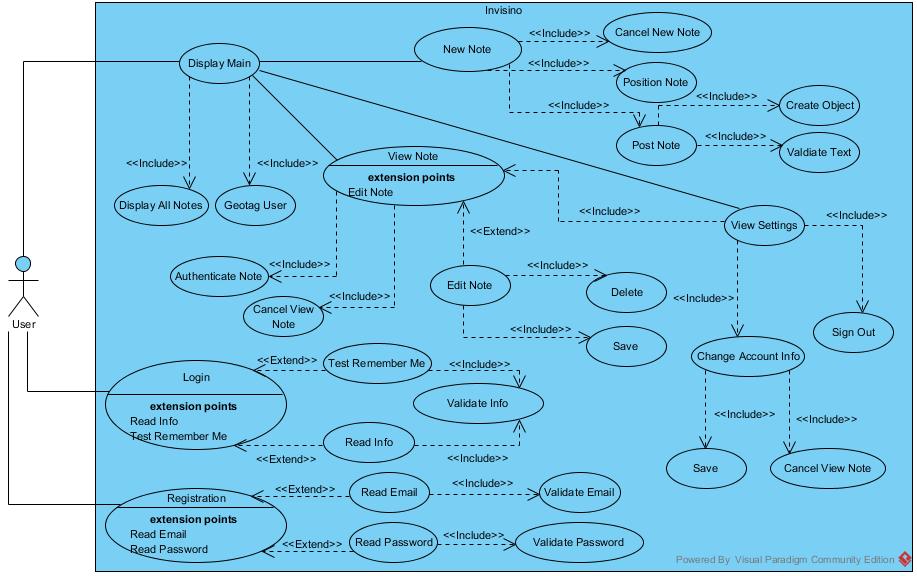
1. **Primary Features**
2. **Secondary Features**

**III Summary of Requirements:**

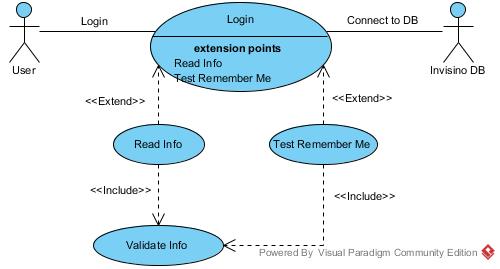
1. **Functional Requirements**
2. **Nonfunctional Requirements**

**IV Diagrams:**

1. **Use Case Diagrams**



The diagram above is the overall use case diagram, and the individual ones follow below. These describe all of the different possible routes that a user of Invisino can go through.



**Use Case:** *Login*

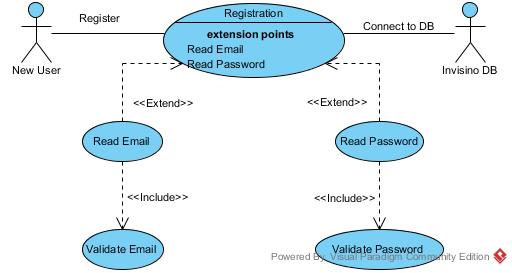
**Context:** The “Login” user case will allow a user to login with their email and password in order to access the rest of the application. If they have already logged in before and checked the Remember Me flag, the app will automatically login for them.

**Actors:** General User

**Main Success Story:**

1. Invisino DB checks if given device has already checked Remember Me flag for a given user, and login for them if it is true.

2. If Remember Me is false, the user will have to login using their credentials. The Invisino DB will authenticate their info and login if they are correct, otherwise prompt user to re-enter their info.



**Use Case:** *Registration*

**Context:** The “Registration” use case allows a new user to sign up an account to use the app. They will have to enter their email and create a password to do so.

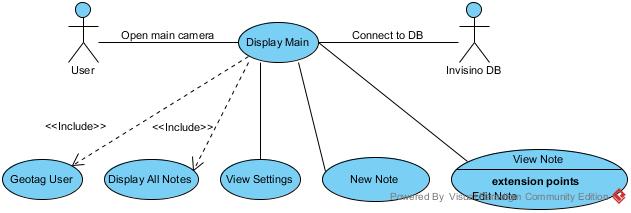
**Actors:** New User

**Main Success Story:**

1. Invisino DB will read the email and password that the new user enters and validate if they are acceptable.

2. If acceptable info has been entered, the user will be redirected to the login page and be logged in.

3. If the info is not acceptable, then they will be prompted to re-enter their info according to the requirements.



**Use Case:** *Display Main*

**Context:** The “Display Main” user case follows immediately after a user has logged in and is the main screen of the application. Here, the user is shown all notes within range to their physical location and has the option to view these notes, create their own note(s), or view their settings.

**Actors:** General User

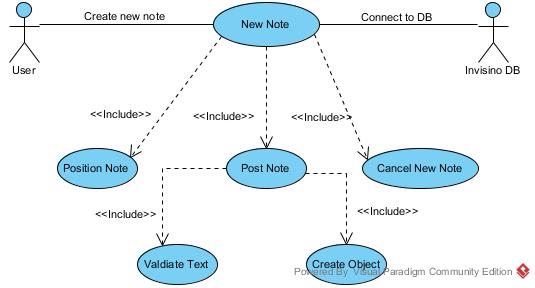
**Main Success Story:**

1. Invisino DB geotags the user’s device and pulls all of the nearby notes in range to display to the user.

2. If the user taps on the button for creating a new note, the “New Note” use case starts.

3. If the user taps on an existing note, the “View Note” use case starts.

4. If the user taps on the settings button, the “View Settings” use case starts.



**Use Case:** *New Note*

**Context:** The “New Note” use case allows a user to create their own note that will be geotagged to their current location. They

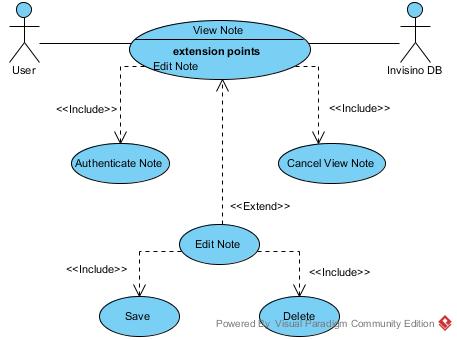
**Actors:** General User

**Main Success Story:**

1. The user will position the note before they add text to it.

2. Once positioned, the user will be prompted to provide text for the note. If they press on the “Post” button, the text will be validated; if the text is ok, then the note will be successfully posted, otherwise the user will be prompted to change their text to match the requirements. Once valid text is entered, Invisino DB will create that note object.

3. If at any point a user wants to cancel making a new note, they simply press cancel and will be returned to the main screen.



**Use Case:** *View Note* (and within, *Edit Note*)

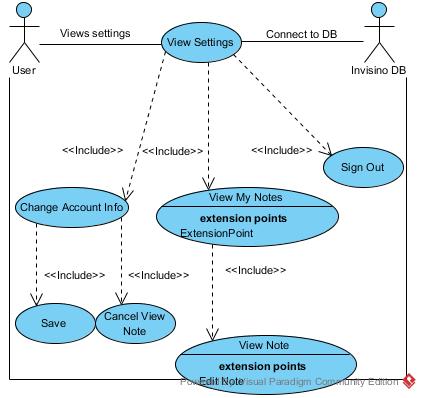
**Context:** The “View Note” user case allows a user to look at an already existing note. If the user is the creator of that given note, then they will be given the option to edit it.

**Actors:** General User

**Main Success Story:**

1. Invisino DB will authenticate if the user is the one made the note. If so, they will be able to open the sub use case “Edit Note”. Within that, they can either save their changes or delete the note entirely.

2. If at any time the user wants to go back to the main screen, they can cancel the view.



**Use Case:** *View Settings*

**Context:** The “View Settings” use case will allow a user to change their account info, view their current notes, or simply sign out. There is always a back button to go back to the main screen as well.

**Actors:** General User

**Main Success Story:**

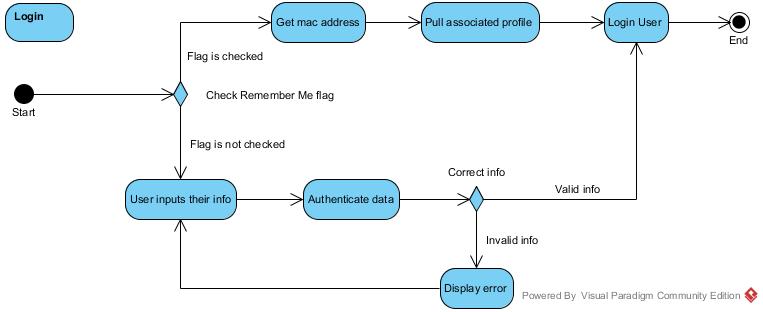
1. If the user wants to change their account info, they can edit their current email and password, so long as the new info is valid.

2. If the user wants to view their notes, they will be shown a list of their current notes. They can view and edit each one of them as wanted.

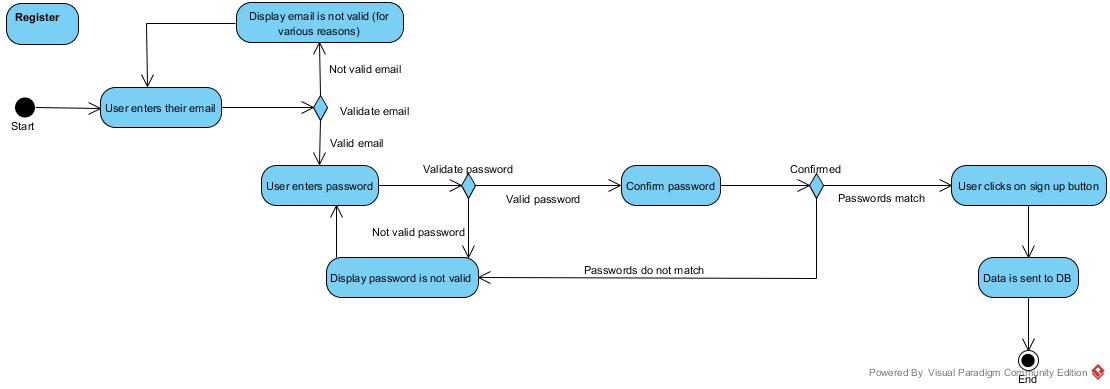
3. The user can press the sign out button if they want to switch accounts or get out of theirs for any reason.

1. **Activity Diagrams**

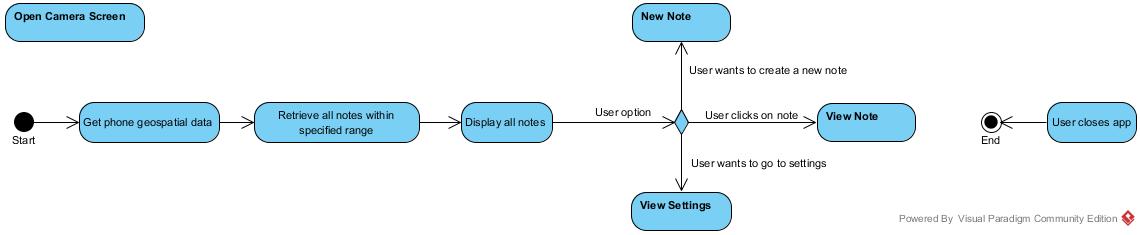
These diagrams demonstrate the flow of actions taken when a user opens and uses the application. When Invisino opens, the first activity is the login activity.

****

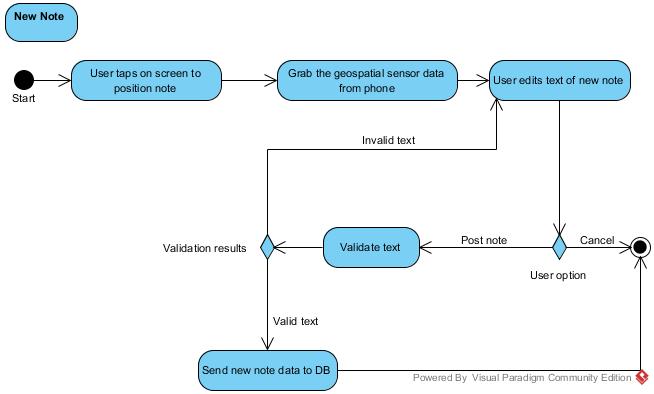
If a person doesn’t have an account, they have to register, which opens the register activity. Otherwise, it goes to the Open Camera Screen activity.



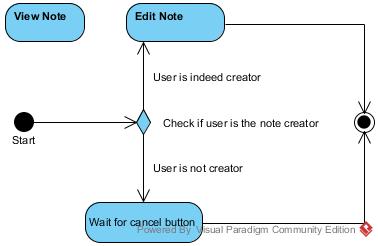
The user then registers, which goes back to the login screen and automatically logs them in. The user is then redirected to the Open Camera Screen activity.



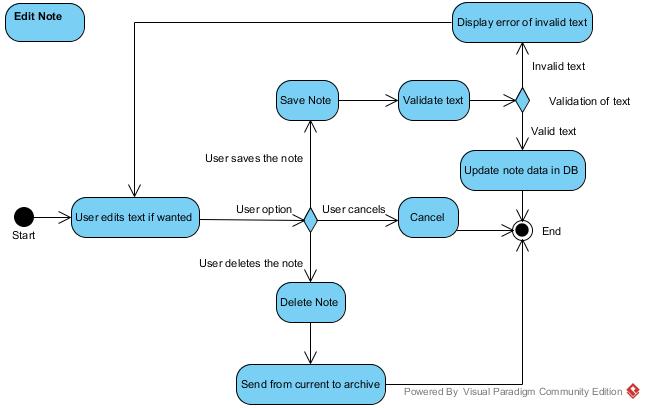
The user then has different options available to them. If they want to make a new note, it opens the New Note activity.



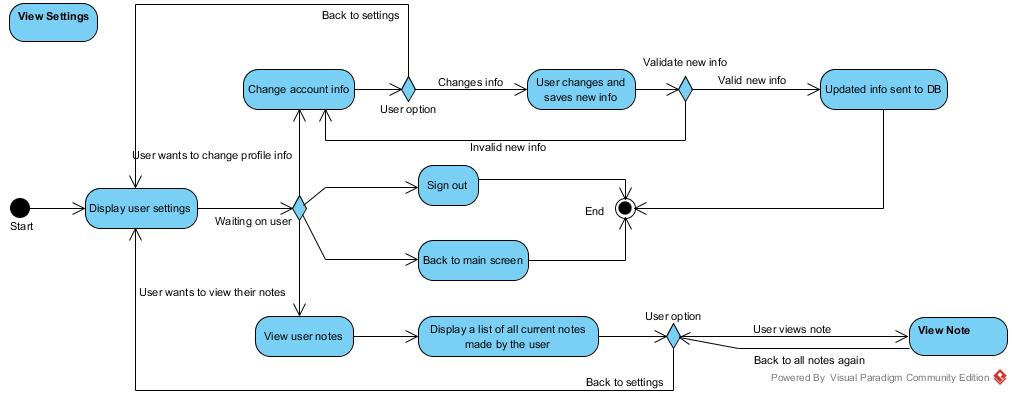
Once the user finishes making the note, they return to the camera screen. There, they also can look at current notes on the screen, the View Note activity starts.



From here, the user can cancel the view if they want to return to the main camera screen. However, they can also edit the note if they are the note’s creator, opening the Edit Note activity.

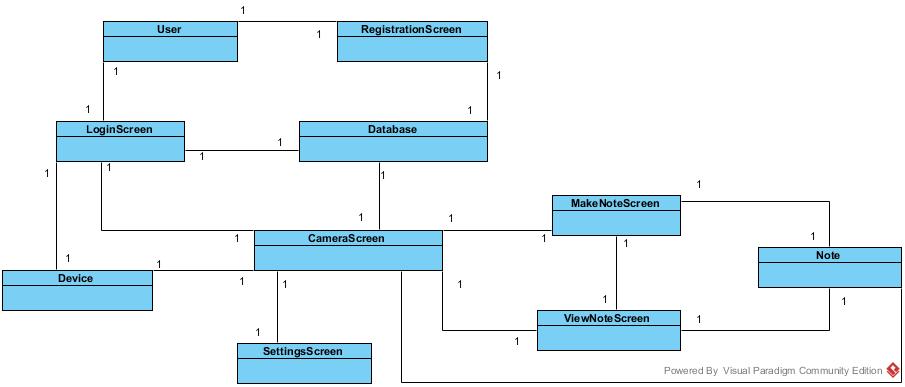


The note creator can edit the text, delete the note, or decide to cancel editing. The user then can go back to the camera screen. On that screen, if the user wants to see their profile, they open the View Settings activity.



Here, the user can change their account information (email and password), view their current notes (which includes View Note activity), or simply cancel and go back to the camera screen.

1. **Class Diagrams**
   1. **High Level**



* 1. **Detailed**

Detailed class diagram to be created in the design phase.

1. **Sequence Diagrams**

No sequence diagrams for the requirements phase, to be continued in the design phase.