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
- Created UI Models – September 19, 2017 Jessie
- Added activity diagrams – September 24, 2017 Maggie
- Added use case diagrams – September 25, 2017 Maggie
- Finalized UI Models – September 25, 2017 Jessie
- Edited descriptions – September 26, 2017 Everybody
- Filled out remaining details – September 27, 2017 Maggie
- Enlarged diagrams – October 3, 2017 Maggie

Table of Contents:

I Introduction:

A. Motivation

This application's motivation began with the desire to provide a more contemporary and renewed method of communication. With the use of augmented reality, our project brings connection into a new realm. The virtual world allows people to experience a fresh

outlook on their surroundings and opens new doors for people to reach out to others. By developing virtual notes overlaying the real-time camera feed, a unique environment for sharing ideas is formed. Tagged to a geographical location, these notes allow users to converse anonymously and explore their ideas freely using AR technology; this empowers people's creativity. As social media and technology as a whole progresses towards the AR and VR age, Invisino will already be there, ready and capable for users to enjoy.

B. Scope

The scope of this project is extremely scalable. Since the notes are geographically tied to a location, Invisino can be used both locally and globally. The app offers varying uses ranging from interactive message passing to exciting scavenger hunts - the possibilities are endless. While all of the notes are currently public, there is the possibility of branching out and having private notes as well as group-specific ones. These two possible features are stretch goals at the moment and will be implemented if time permits.

C. Goals

The project's goal is closely tied to the motivation itself. While Invisino doesn't solve an issue, it instead provides something new to the social market. The base requirement is to connect multiple devices to a cloud-ready database, with efficiency and no memory leaks. From there, our initial goals are to make a simple, efficient, and well-rounded application that users can operate with ease. These can be extrapolated upon to stretch goals such as allow different note objects/shapes. Moreover, client-side timing capabilities for notes are also probable options.

D. Relevant Definitions

AR (Augmented Reality) – technology that superimposes a computer-generated image onto a user's view of real surroundings
AWS – Amazon Web Services
VR (Virtual Reality) – computer-generated simulation with 3D environment

II Project Description:

We will be creating an Android application that allows users to experience a new way of conversing. The focal component of this app will be utilizing the user's GPS to track their physical location and display all the notes within range around them. As the person moves, their surrounding notes will be recalculated and presented accordingly. Individuals will also have the ability to create their own notes and view ones that are already in existence nearby.

A. Primary Features

These features are the main goals of the application that are considered a requirement for the application to be completed. They are as follows:

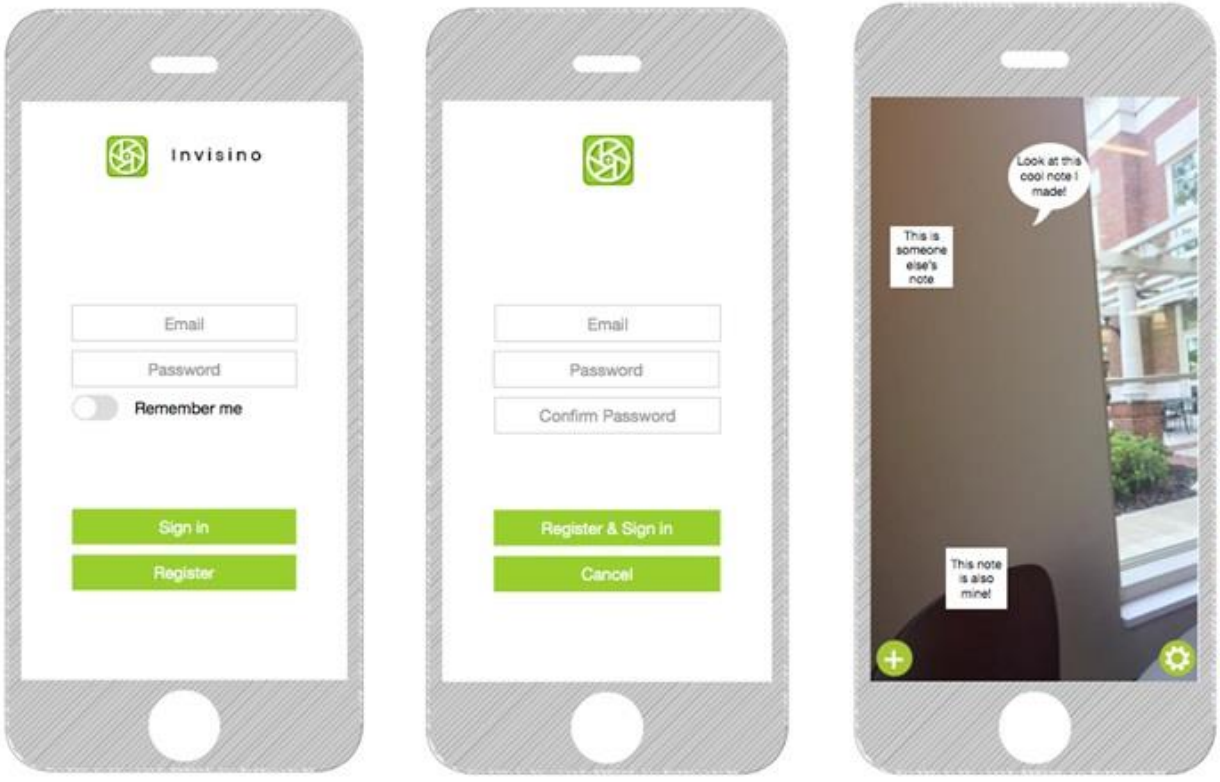
1. Display all notes in the area that are within range.
2. Create a note and have it posted to the user's location.
3. View notes that are nearby by tapping on them.
4. Edit notes if and only if the user is the one who created them.

B. Secondary Features

These features are not required to complete the application and are stretch goals that can be added only once the primary ones have been completed.

1. User-edited time limit on notes rather than it being server-side controlled.
2. Forgot my password option on the login screen that aids users.
3. Different types of note shapes and textures, like balloons and different colors.
4. Add privacy capabilities, allowing public and private notes.
5. Turn on/off profanity to show or hide notes around the user.
6. Block users in case of someone misusing the application.
7. Create a map of all notes within the area.

C. UI Prototype Examples



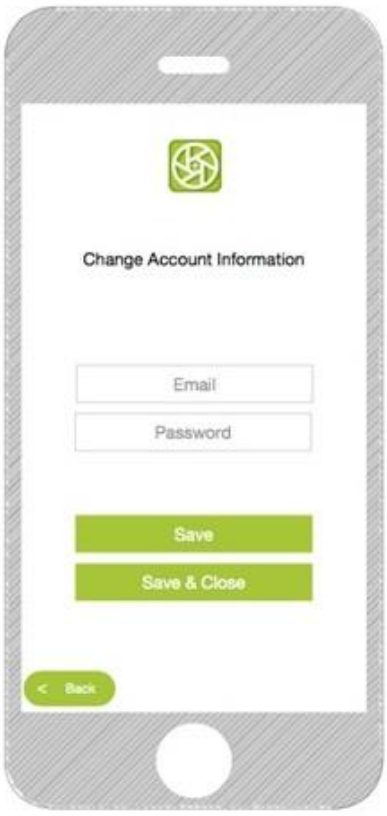
Login Register Main Camera

New Note

View (Your) Note

View (Others) Note

Settings



Change Account Info

View My Notes

III Summary of Requirements:

A. Functional Requirements

These are all high priority tasks that are necessary for the success of this application.

1. Display All Notes
   Pulls a list of all nearby notes from the database and displays them accordingly.
2. Create Note
   Allows a user to create their own note tied to their current location pulled from their device's GPS.
3. View Note
   Displays a selected note full screen to be read (this is extended upon with the edit note capability).
4. Edit Note
   Permits a user to edit one of their own notes, by changing the note text or simply deleting it altogether.
5. Settings
   Shows the user's profile and gives them options to see all of their current notes, change account info, or sign out.

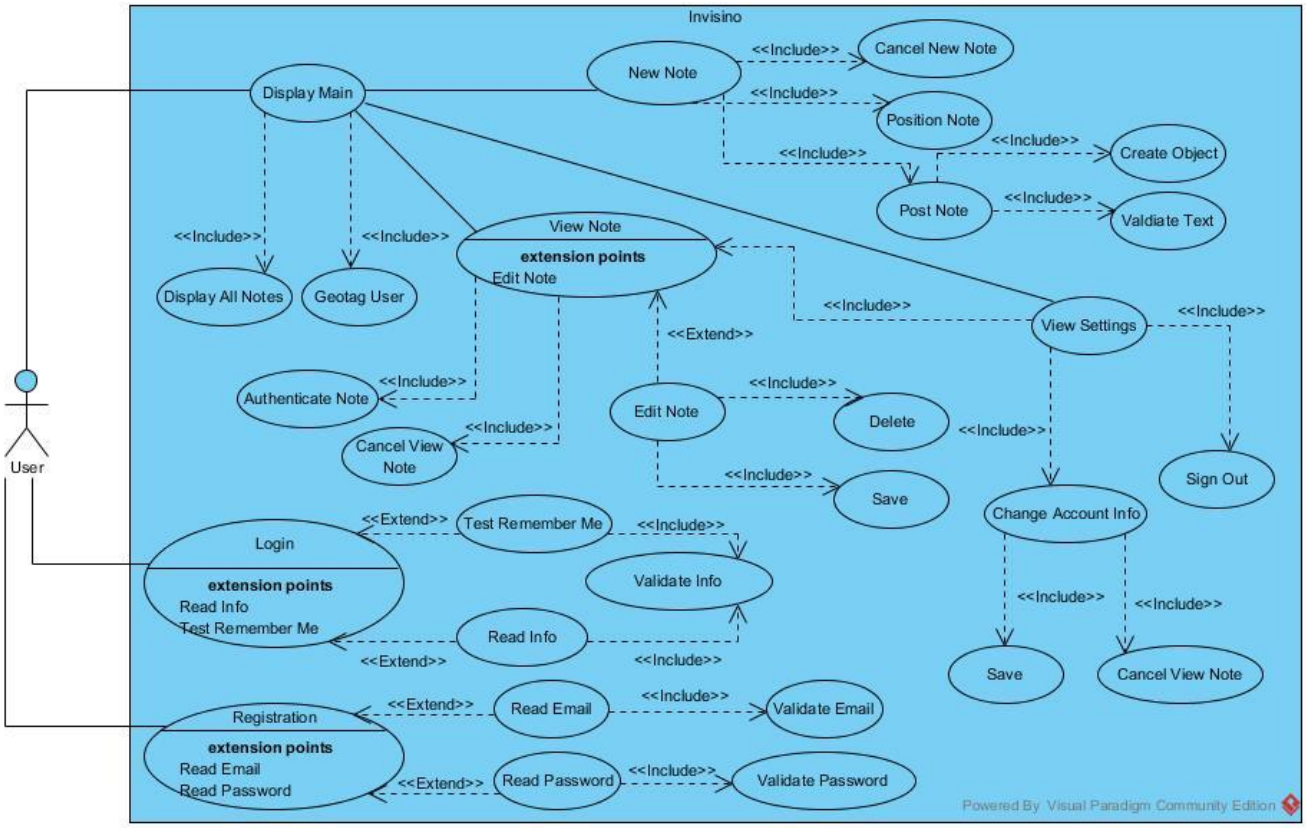
B. Nonfunctional Requirements

These are quality attributes that are not essential but provide more functionality to the application and are ranked from highest priority to lowest.

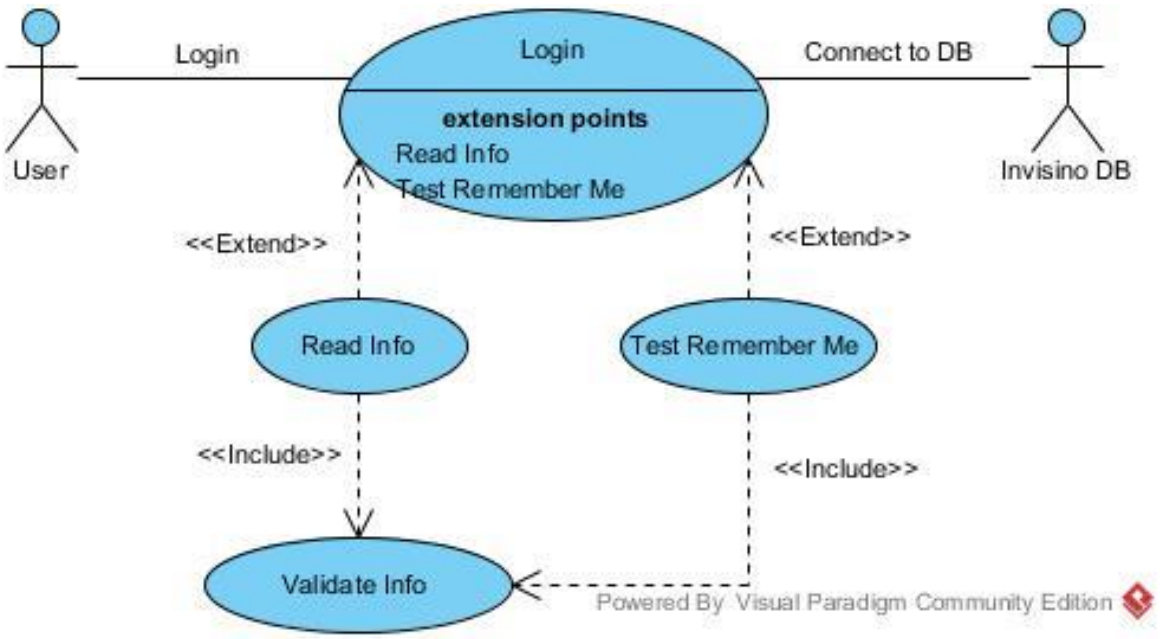
1. Unique Logins
   Users will have their own accounts so that their notes are editable only by themselves as well as stored accordingly, allowing for systematic retrieval from the database.
2. Efficiency
   The app must run quickly and adeptly regardless of the number of devices all connecting to the cloud database, while not taking up too much space on each of these devices as well.
3. Ease of Use
   Since there is the possibility of a wide range of users, the app must be intuitive and sleek.
4. Scalability
   Since the application is not restricted and uses users' GPS, it can be utilized anywhere and must also allow for future features to be built in at any point.

IV Diagrams:

A. 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. The main three options are to login, register, and display the main screen once the user has signed in with their account. From the main screen, there are several more subsequent options that include viewing existent notes within range, creating a new note, and viewing one's settings. If a user views a note that they made, they will be able to edit the text within the note or delete it altogether. If the user goes to their settings, they have the ability to view (and edit) all of their current notes, change their account information, or simply sign out. The descriptions of each use case starts here:

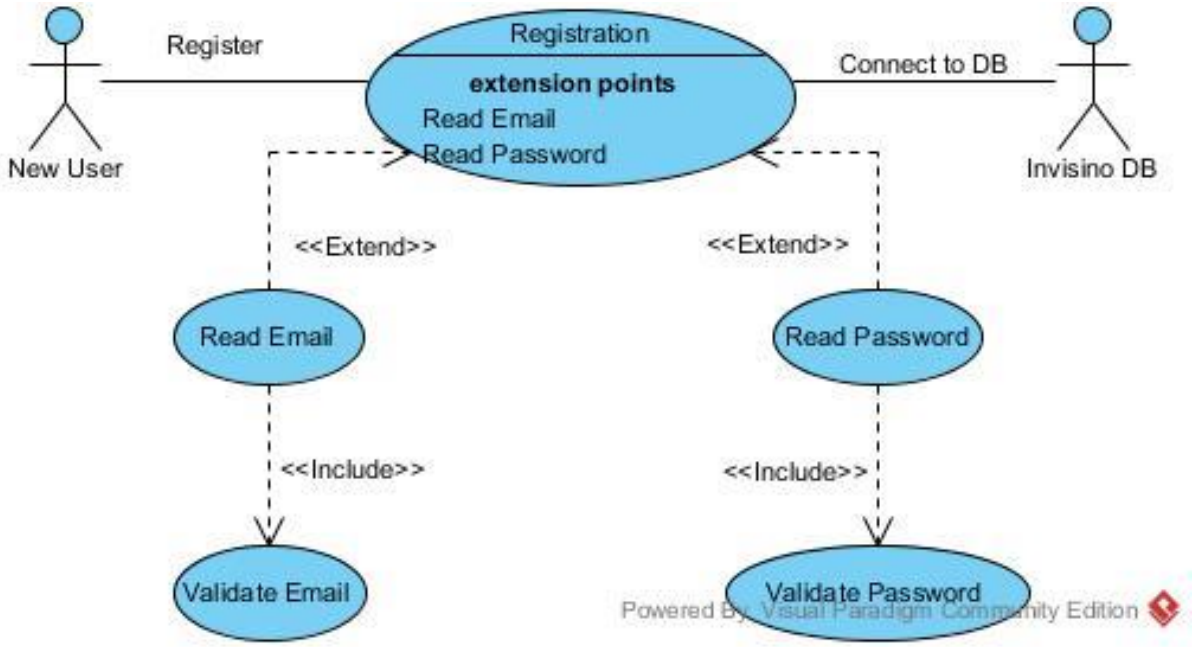


Use Case: *Login*

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.

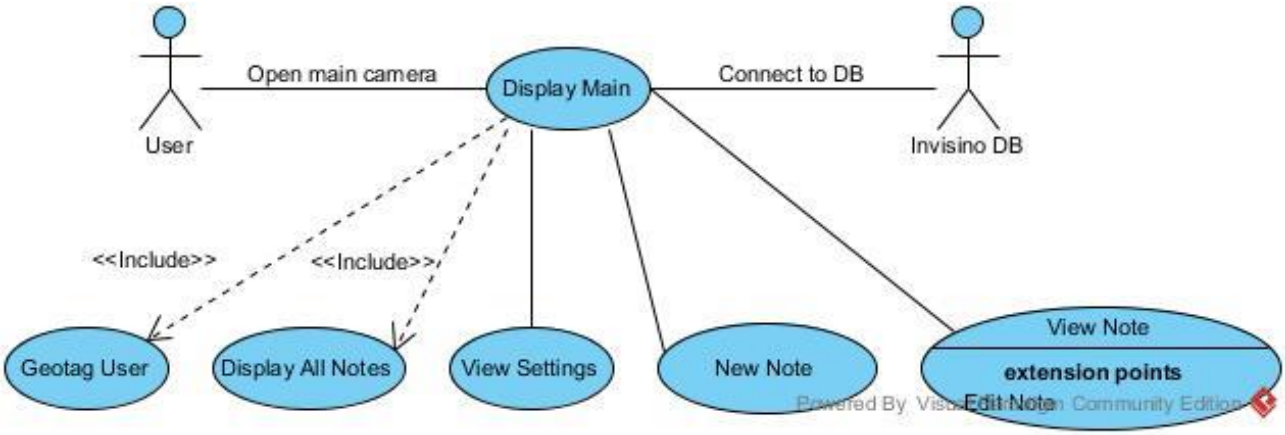


Use Case: *Registration*

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.

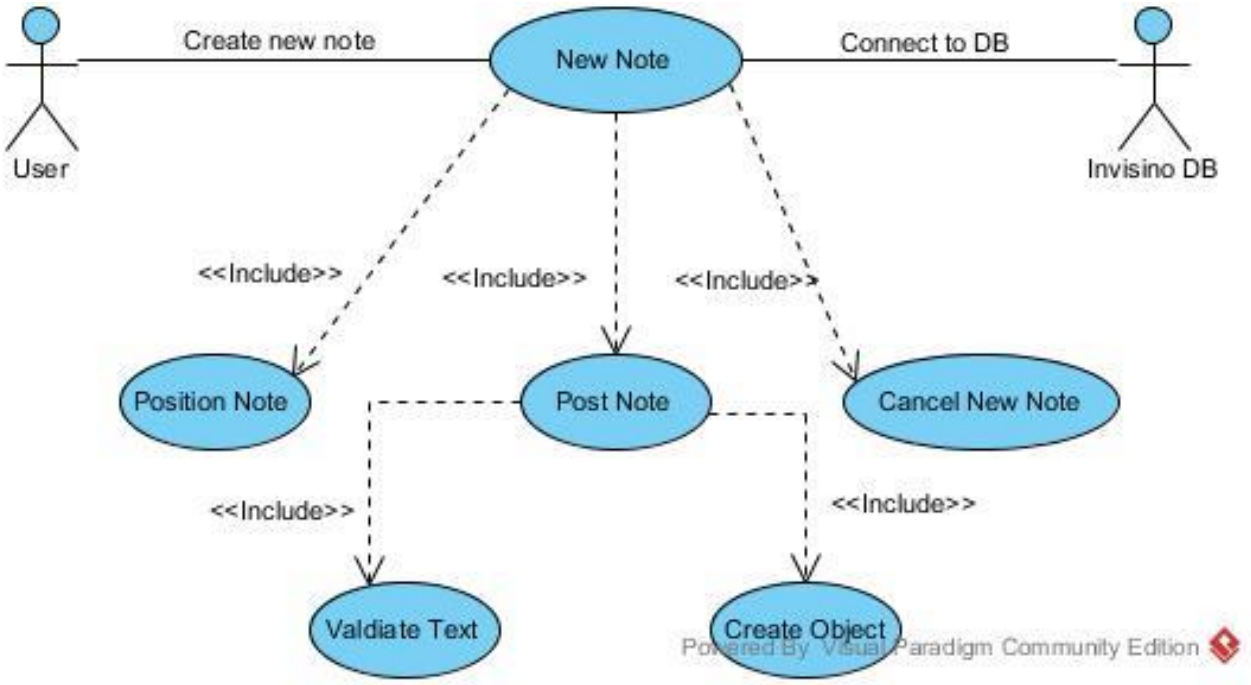


Use Case: *Display Main*

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 new note button, 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.

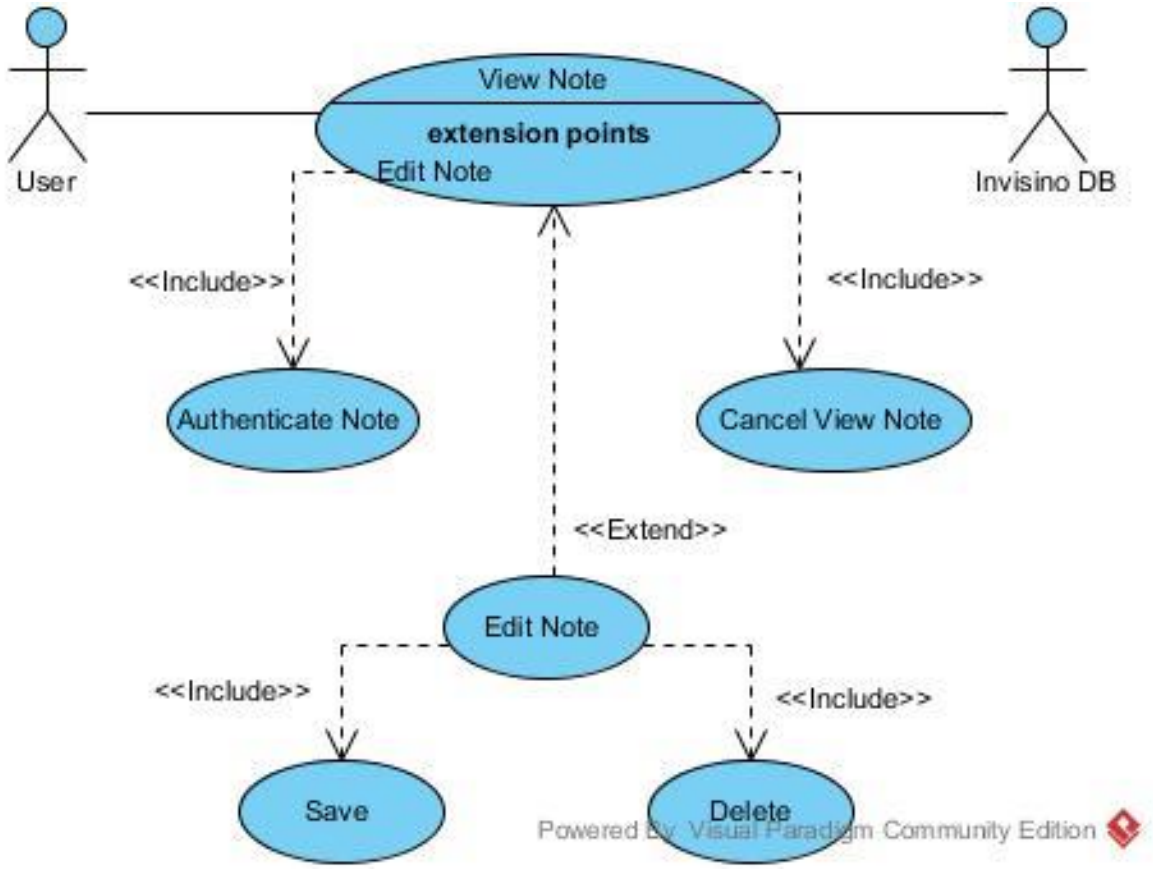


Use Case: *New Note*

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.

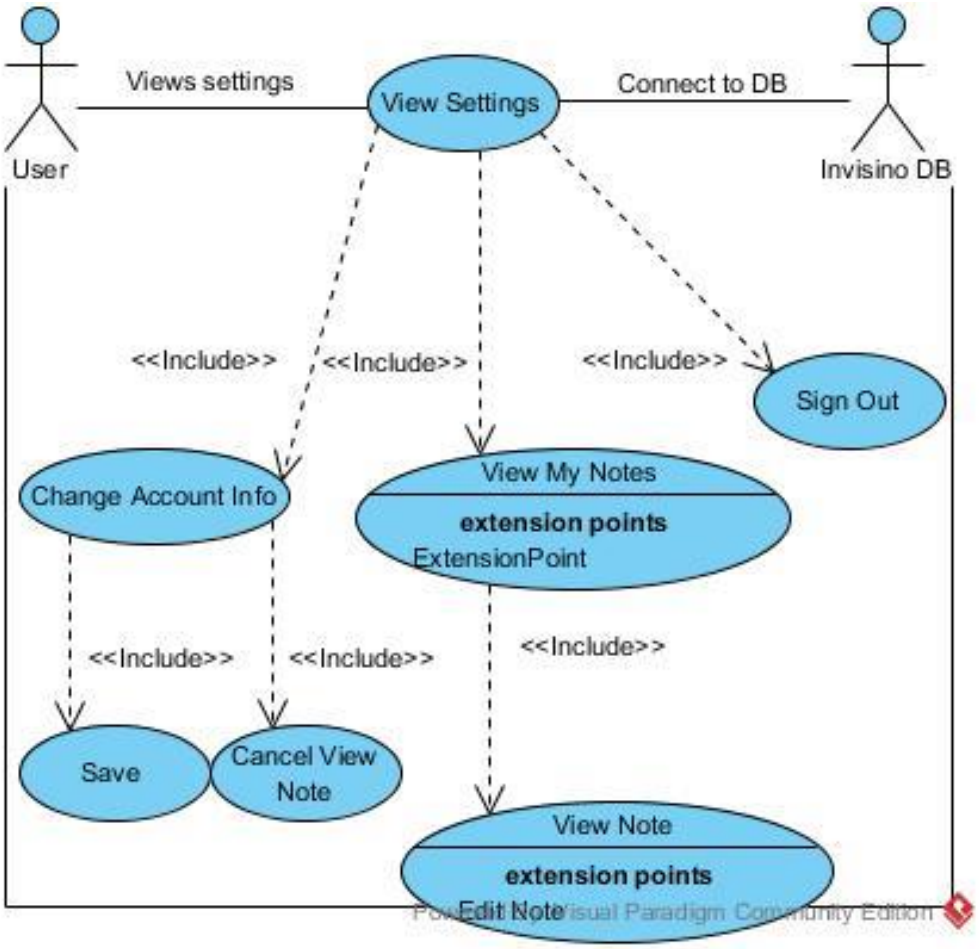


Use Case: *View Note* (and within, *Edit Note*)

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*

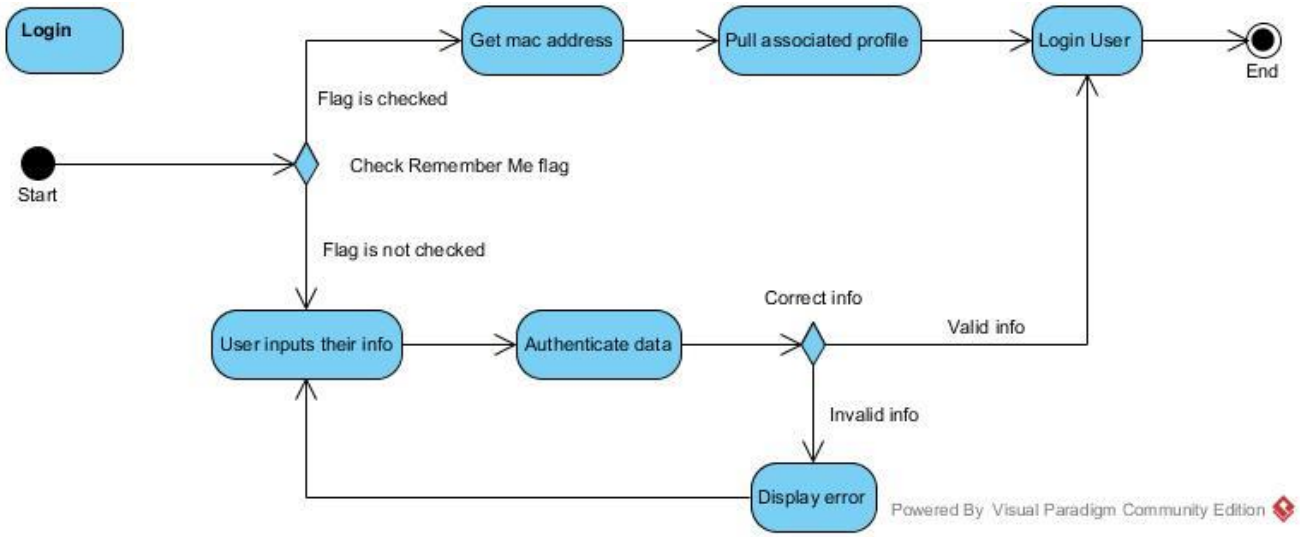
Context: The "View Settings" use case will allow a user to change their account info, view their current notes, or simply sign out. There is always a back button to go back to the main screen as well.

Actors: General User

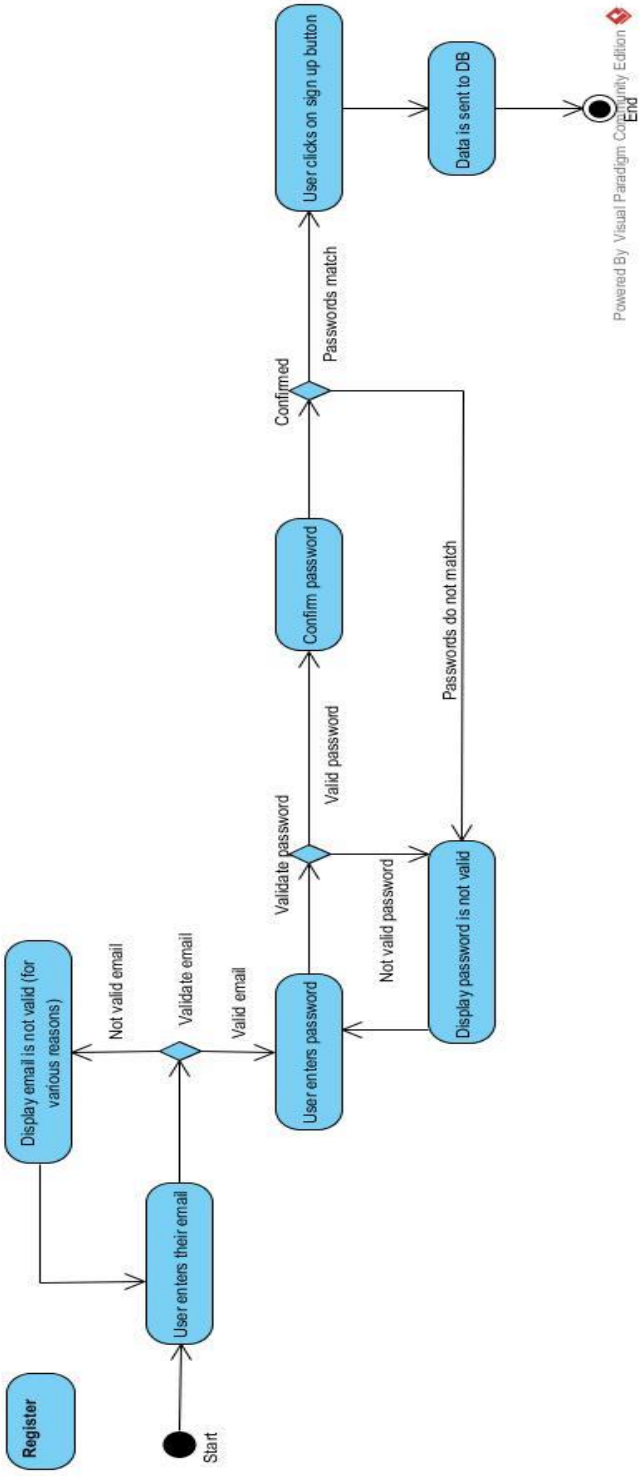
Main Success Story:
1. If the user wants to change their account info, they can edit their current email and password, so long as the new info is valid.
2. If the user wants to view their notes, they will be shown a list of their current notes. They can view and edit each one of them as wanted.
3. The user can press the sign out button if they want to switch accounts or get out of theirs for any reason.

B. Activity Diagrams

These diagrams demonstrate the flow of actions that can be taken when a user opens and uses the application. At any point, the user can close the application, ending the currently running activity. When Invisino opens, the first activity is the login activity.

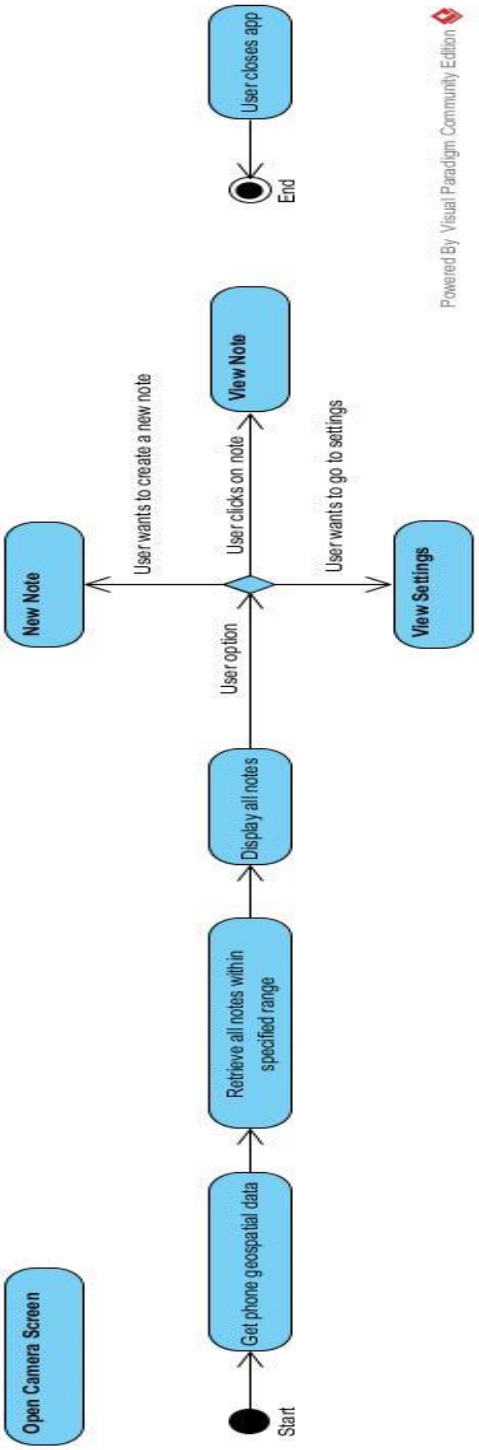


Invisino will confer with the backend database to see if the user has already logged in before and asked to be remembered. If the person has, the database will get the mac address and auto-login the user for them. If they have not checked the Remember Me flag, the person will have to input their information correctly to login. If a person doesn't have an account, they will have to sign up using the respective button, which opens the register activity. Otherwise, if a person logs in successfully, Invisino goes to the Open Camera Screen activity.

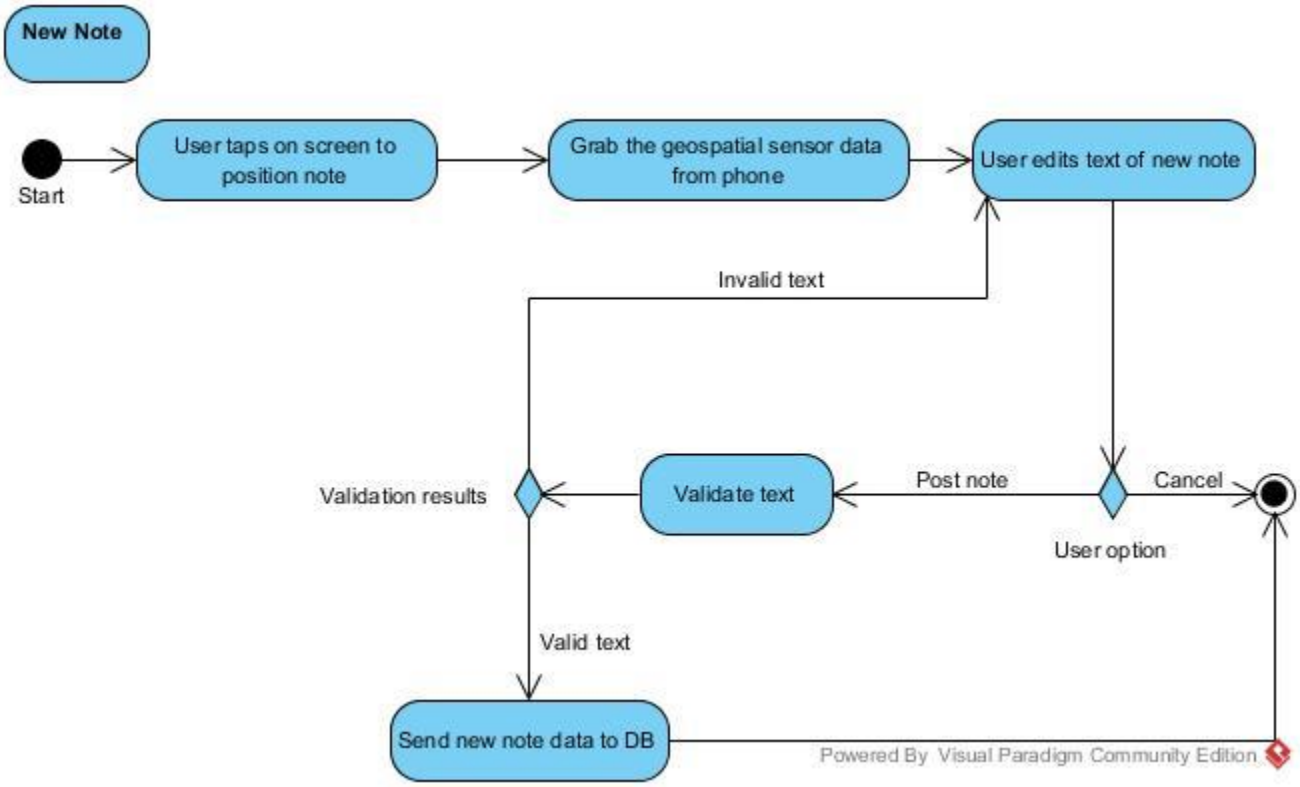


In the case that the user must register, they are prompted to enter their email and password, both of which must be validated, and also confirmed in the case of the password. The user will be notified if either of them are not valid and will have to change them if that is the case. Otherwise, the information they entered is valid. In that case,

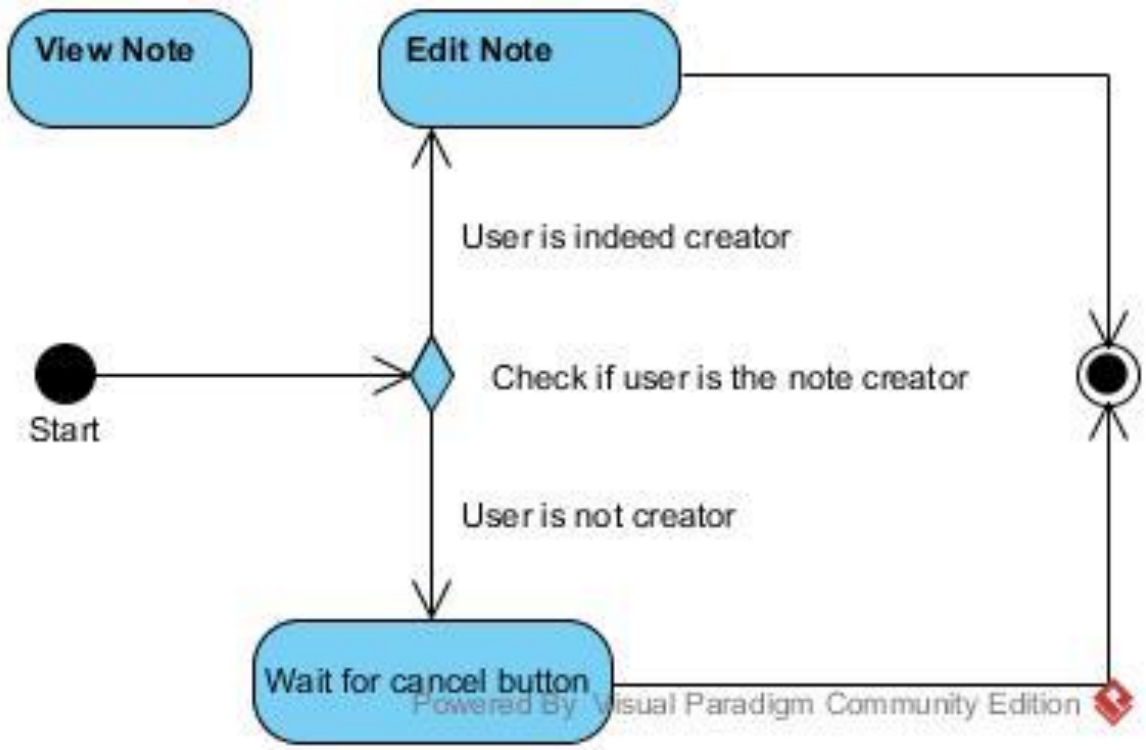
once the user clicks the register button, the information is sent to the backend database, and the user is now registered. Invisino now goes back to the login screen and automatically logs them in, redirecting again now to the Open Camera Screen activity.



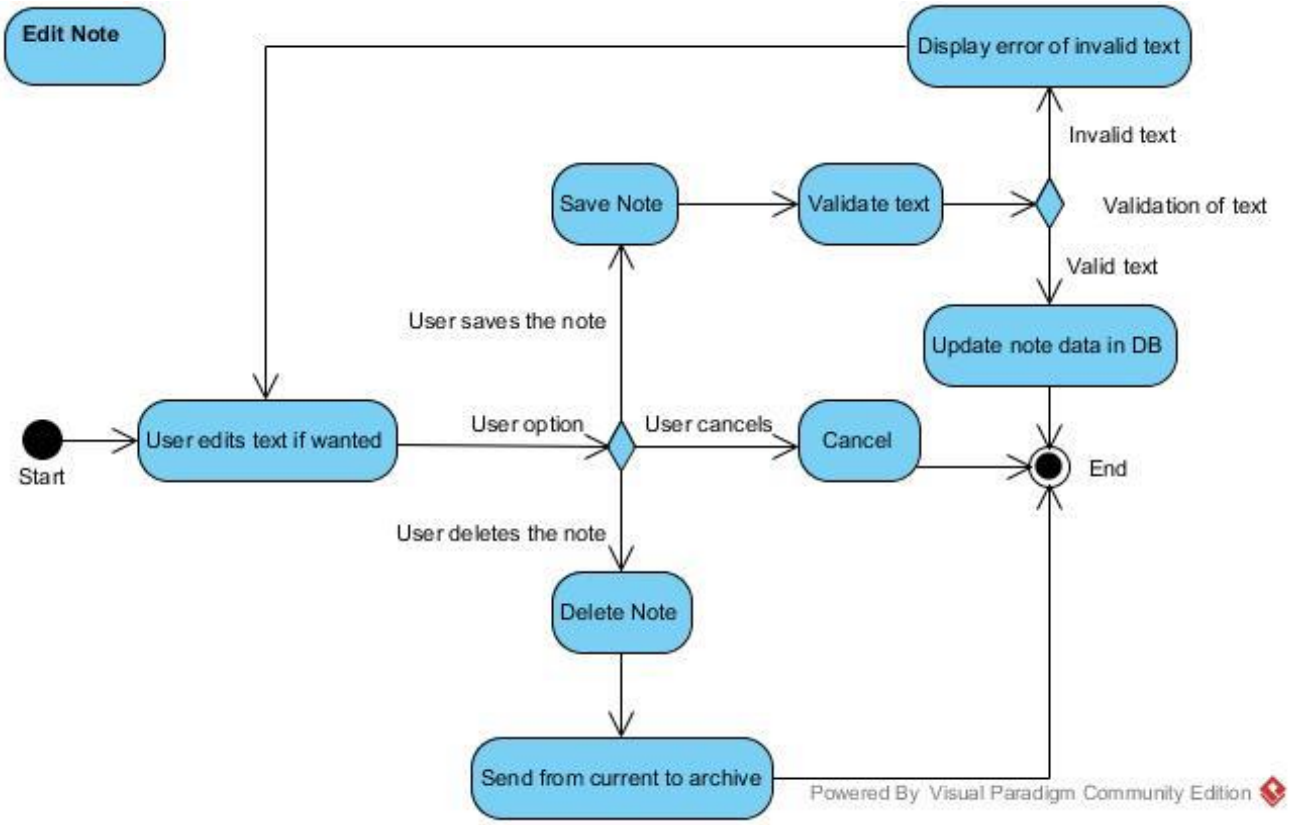
This use case is the main camera screen of the application. Here, the backend database is in routine communication with the device in order to track its location. This way, any and all notes within range of the phone will be picked up and displayed to the user. Moreover, the user has different options available to them. Once again, if the user closes the app, the activity ends and the app stops. If they want to make a new note, it opens the New Note activity.



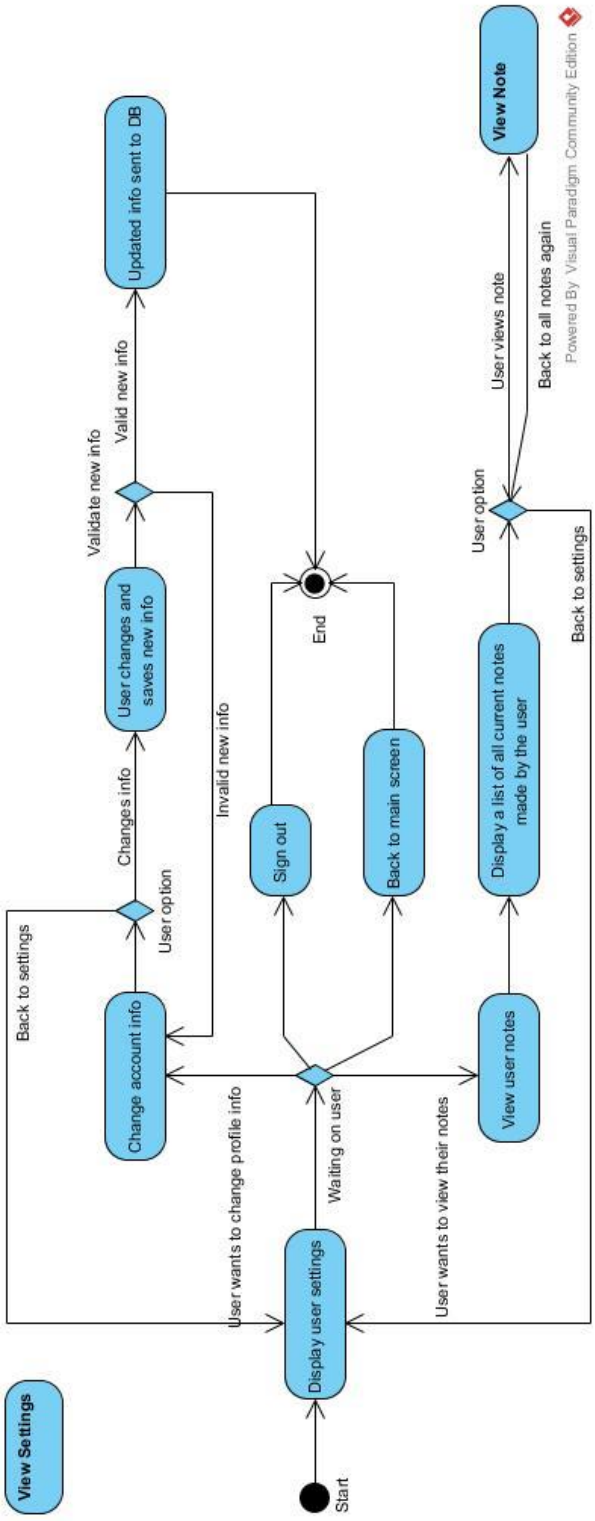
To make a new note, the user must position the note first. Once the note has been put into its wanted location, the Invisino database picks up the phone's geospatial data to add to the note information. Then, the user fills out text for the note, which must be validated before it can be posted. Once the user finishes making the note, the note data is sent to the database. Then, Invisino returns to the camera screen. There, users also can look at current notes on the screen, when the View Note activity starts.



This use case opens up the note in full screen view for the user to read. 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.



Here, the note creator can edit the text, delete the note, or decide to cancel editing. If they edit the text, it must be validated before the note can be saved. Otherwise, the user will be notified that they must provide text that fits the requirements. Only once valid text has been entered will the note data in the database will be updated. If they want to delete the note at any time, the note will be sent from current notes to the archive and will be no longer visible or even accessible. The user then goes back to the camera screen. On that main screen, if the user wants to see their profile, they open the View Settings activity.

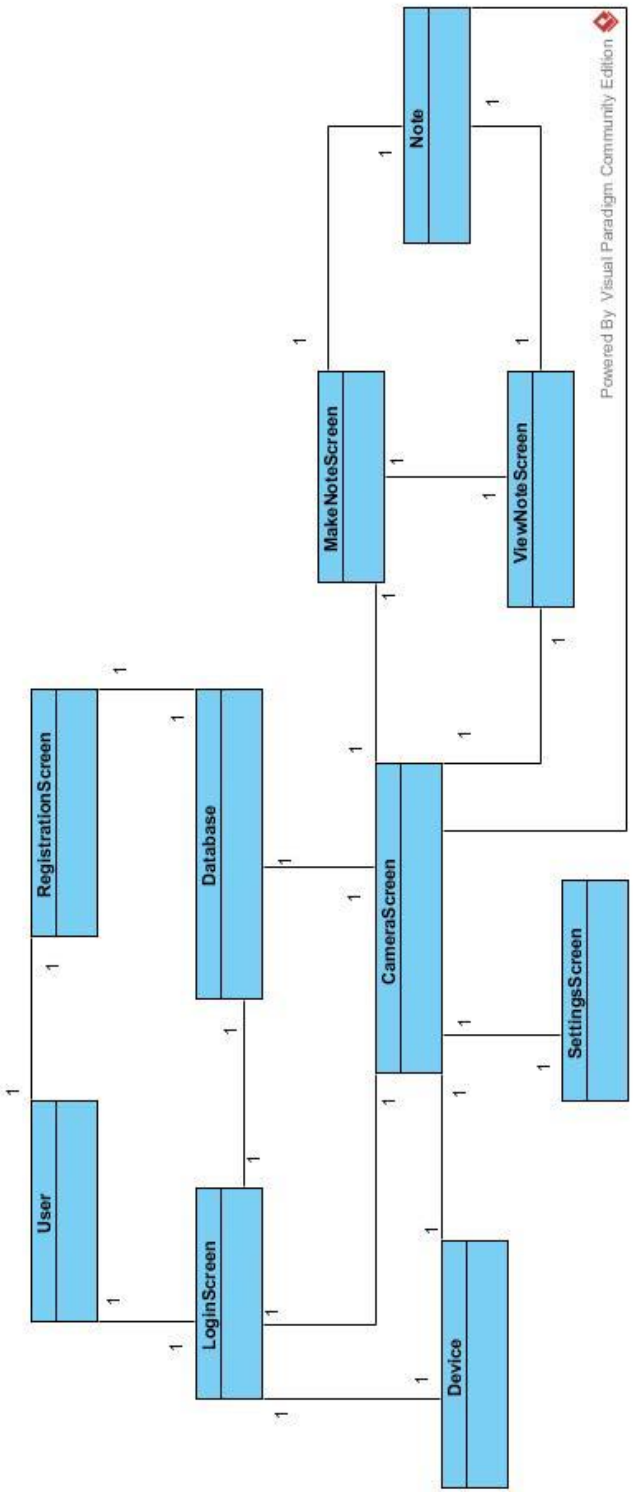


On the settings/profile page, the user has several options. If they want to edit their account info (email and/or password), whatever they change must be validated before it is sent to the database. If they want to view all of their own notes, a list of their currently

existing notes will be displayed. The user can view (and edit) any of these notes as wanted. At any point during these options, they can also hit the back button to get back to the main settings screen. If they want to sign out, they are automatically redirected to the login screen because they are no longer in their account. There is also a back button on settings to return to the main camera screen.

C. Class Diagrams

i. High Level



The above diagram is the high level class diagram for Invisino. One important thing to note is that any line with no multiplicities is automatically assumed to be a 1-to-1 relationship. Any other multiplicities are explicitly defined.

ii. Detailed

Detailed class diagram to be created in the design phase.

D. Sequence Diagrams

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