Software Requirements Specification

What's HappNin

Lauren Finley, David Lucero, Sumaiya Rashid,
Yavuz Kocoglu,Rehan Ayoub, Andres Suarez, Zack Luckman

Bashes

May 6, 2018

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification document is to describe, in detail, the functionalities of What's HappNin, a location based social media application. This document will explain the purpose, features, constraints, interfaces, and functionality of What's HappNin.

1.2 Document Conventions

This document was written using the IEEE template for System Requirements Specification.

1.3 Scope

The purpose of What's HappNin (WHN) is to allow users to connect with their friends by sharing the activities that they are doing. The users can achieve this by utilizing a map which pinpoints the locations where all of the user's friends are gathered and partaking in a social activity. When the user opens the app, they have the option to create an activity themselves by making a series of posts which indicate what kind of social activity the user is partaking in. Furthermore, the user will have the ability to upload comments, pictures or videos to keep their friends updated on the status of the event and entice friends to join. This will be accomplished on a map by including small features such as event icons and notifications that notify other users when posts are being made and of the location in which the activities are occurring. More specifically, the event icons on the map will include a symbol such as a basketball to let their friends know what type of event is taking place, the number of people of attending the event, and if clicked the event icon will redirect the user to the event feed. From this feed WHN users can see photos, comments, and videos that have been posted to an event. There will also be a point system in place to award users for creating or attending social events with their friends.

1.4 Definitions, Acronyms, and Abbreviations

1.4.1. Definitions

Hap: A picture, video, or comment posted to the feed of an event

1.4.2. Abbreviations

WHN: What's HappNin
FR: Functional Requirement
NFR: Non-Functional Requirement
GPS: Global Positioning System
UC: Use Case

1.5 References

Google Maps APIs: https://developers.google.com/maps/

1.6 Overview

Part 1 of this document consists of a brief introduction to the What's HappNin application. Readers seeking a more detailed description should refer to part 2 of the document, which describes the functionality and characteristics of the application. Part 3 contains the system requirements, user requirements, and constraints of the application. References to each section can be found in the Table of Contents.

2. Overall Description

2.1 Product Perspective

What's HappNin is a new, self-contained product that does not rely on or replace currently existing applications. What's HappNin is designed for anyone who is interested in connecting instantaneously with their friends based on their real-time location and activities that are currently taking place. The product shall exist in the form of a smartphone application.

2.1.1. System Interfaces

This system will interface with the Android mobile operating system on the device and a client-server database will be used to store user information.

2.1.2. User Interfaces

The user interface will follow Android Material design standards. See section 4 of this document for more information.

2.1.3. Hardware Interfaces

There are no specific hardware requirements if the software requirements are met.

2.1.4. Software Interfaces

On the user's Android smartphone, an operating system of version 4.4 (KitKat) at the oldest is required. The client-server database will be a MySQL database that stores user information and configuration options.

2.2 Product Functions

The primary functions of What's HappNin are as follows:

- Create an event
- View the location of an event on a map environment

- Check in to an event
- Upload media to the feed of an event
    - Pictures
    - Videos
    - Comments
- See which of your friends are checked in event

2.3 Design and Implementation Constraints

Constraints to the system include the following:

- The user must have an Android smartphone capable of running Android 4.4 (KitKat) and above.
- The smartphone must be charged and functioning.
- The smartphone must have GPS capabilities that are enabled.
- The smartphone must have cellular or wifi data enabled.
- The smartphone must be within a coverage area provided by the user's data plan.
- All permissions for the app are granted.

2.4 Assumptions and Dependencies

The following assumptions are made about the users of this application:

- The user has the downloaded WHN onto their smartphone.
- The user has a smartphone operating on the Android mobile operating system.
- The user has a data plan for their smartphone to be able to receive data coverage.
- The user has enough storage on their device to support the application.

3. Specific Requirements

3.1 Use Cases

3.1.1. Sign Up

Description: Unregistered user makes an account
User Story(s): 2,11,9
Actors: User, Profile Database, Authentication Database, Application
Preconditions: User is not logged in and has never made a profile
User steps:

1. Starts up the application
2. Chooses option to sign up
3. Types in their wanted username and password
4. Accepts terms and conditions agreement

System steps:

1. Checks their username against the Profile database to see if it is already taken. If not, it will then create a profile associated with it in the Profile database.

Post-conditions: User will be logged in as a registered user.
Exceptions handled: If their requested username is already taken it will prompt them with this error. If they do not follow the rules for the password it will prompt them to.
Following Use Cases: <<Extends>> Sign In.

3.1.2. Sign in

Description: registered user is trying to access the application
User story(s): 2,11
Actors: User, Authentication Database, Application
Preconditions: User is not logged in and has previously made a profile.
User steps:

1. Starts up the application
2. Chooses option to sign in
3. Types in their username and password

System steps:

1. Checks their username and password combinations against the Profile database to see if it is correct.
2. If so, it will then allow the user access to the application.

Post-conditions: User will be logged into their account.
Exceptions handled: If they do not type in an existing or correct username and password combination, it will prompt them with this error.
Following Use Cases: None.

3.1.3. Friend Request (Send)

Description: Signed in user sending friend request
User story(s): 16
Actors: User, Profile Database, Friends database, Application
Preconditions: User is not already friends with requested person.
User steps:

1. Starts up the application
2. Chooses option to add a friend
3. Types in their friend's username.

System steps:

1. Checks entered username against the Profile database to see if it is there.
2. If so, it will send a friend request to the requested person.

Post-conditions: The requested person will now be allowed to accept or decline the friend request.
Exceptions handled: If their requested username does not exist, it will prompt them with this error.
Following Use Cases: None.

3.1.4. Friend Request (Receiving)

Description: Signed in user receiving friend request.
User story(s): 16
Actors: User, Profile Database, Friends database, Application
Preconditions: User is not already friends with requested person.
User steps:

1. Starts up the application
2. Chooses option to add a friend
3. Checked under received requests
4. Accepts or declines friend request.

System steps:

1. If the user selects accept:
    a. The user that requested will not be added to their friends list and they will be added to the friends list of the person that sent the request.
    b. Then the notification will be deleted.
2. If the user declines:
    a. The notification will be deleted.

Post-conditions: The user will now have them either added to their friends list or not depending on their choice.
Exceptions handled: None.
Following Use Cases: None.

3.1.5. Follow a friend

Description: Signed in user wants to follow their friend to be notified when their friend posts something or checks into a gathering
User story(s): 15
Actors: User, Friends Database, Application.
Preconditions: User has requested person added to their friends list and has not followed them yet.
User steps:

1. Starts up the application
2. Chooses to view friends
3. Finds their friend
4. Views their friend's profile

5. Selects to follow said person.

System steps:

1. Marks their friend with a following tag in the Friends Database.

Post-conditions: User will now be notified when this friend posts or checks in somewhere.
Exceptions handled: None.
Following Use Cases: <<Extends>> Send Friend Request.

3.1.6. Post Hap

Description: Signed in User wants to post a notification as to what they are doing
User story(s): 4,5,15,19
Actors: User, Location Services, Posts Database, Application.
Preconditions: User has an account and has already signed in.
User steps:

1. Starts up the application
2. Types in the text box a short description about their event
3. Chooses duration of the event
4. Selects to post event.

System steps:

1. Pulls the text, duration, and location (from location services)
2. Saves the information to the Posts Database.

Post-conditions: The User's hap will now be posted to their friends' maps and feeds for the duration that they chose along with a notification will be sent out to anyone following them.
Exceptions handled: If location services are not turned on, system will request that they must be turned on to post.
Following Use Cases: None.

3.1.7. Delete a hap

Description: Signed in User wants to delete a current hap that they have posted
User story(s): 4,19
Actors: User, Posts Database, Application.
Preconditions: User has an account and has already signed in and a hap that still has time duration left.
User steps:

1. Starts up the application
2. Selects to delete current Hap.

System steps:

1. System deletes their current hap from the Posts Database.

Post-conditions: The User's hap will no longer be posted to their friends maps and feed.
Exceptions handled: <<Extends>> Post a Hap.

Following Use Cases: None.

3.1.8. View Map

Description: Signed in User wants to view their map to see if their friends have posted or checked into anything
User story(s): 4,19
Actors: User, Location Services, Posts Database, Friends Database, Application.
Preconditions: User has an account and has already signed in.
User steps:

1. Starts up the application
2. Selects to view map.

System steps:

1. System displays the map

Post-conditions: The User may now view and select the event icons to show them the information about the events.
Exceptions handled: If the user has no friends that have a hap posted or have not checked into an event, the system will display this information to them.
Following Use Cases: None.

3.1.9. View Hap

Description: Signed in User wants to view the Hap (event feed) of an event.
User story(s): 1, 3
Actors: User, Profile Database, Application.
Preconditions: User has an account and has already signed in.
User steps:

1. Starts up the application
2. Selects to view map
3. Selects an event on the map
4. User can view event comments, pictures, and videos posted about the event in the event feed

System steps: Pulls the event information from Posts Database, displays this to the user in the format of a feed.
Post-conditions: The User may now view the Hap (event feed) and all of it's associated media
Exceptions handled: <<Extends>> View Map.
Following Use Cases: None.

3.1.10. Check into event

Description: Signed in User wants to check into an event that their friend has posted.View the map - Signed in User wants to view their map to see if their friends have posted or checked into anything.

User story(s): 4,19
Actors: User, Location Services, Posts Database, Friends Database, Application.
Preconditions: User has an account and has already signed in.
User steps:

1. Selects to accept the notification that allows them to check into their friends event.

System steps:

1. Checks their friends list, crosses it with the Posts Database to check if a friend has posted with a duration that is still lasting
2. Checks user's location and sees if it crosses with any locations of their friend's current Haps
3. (if so) Prompts the user with a notification that when accepted, it will check them into the event
4. (if so) Will display on the map an icon containing the Hap's information.

Post-conditions: The User's friends may now see the event on the map (if they aren't friends with the original poster of the event).
Exceptions handled: If the user does not have location services turned on, it will not allow them this service.
Following Use Cases: <<Extends>> View Hap.

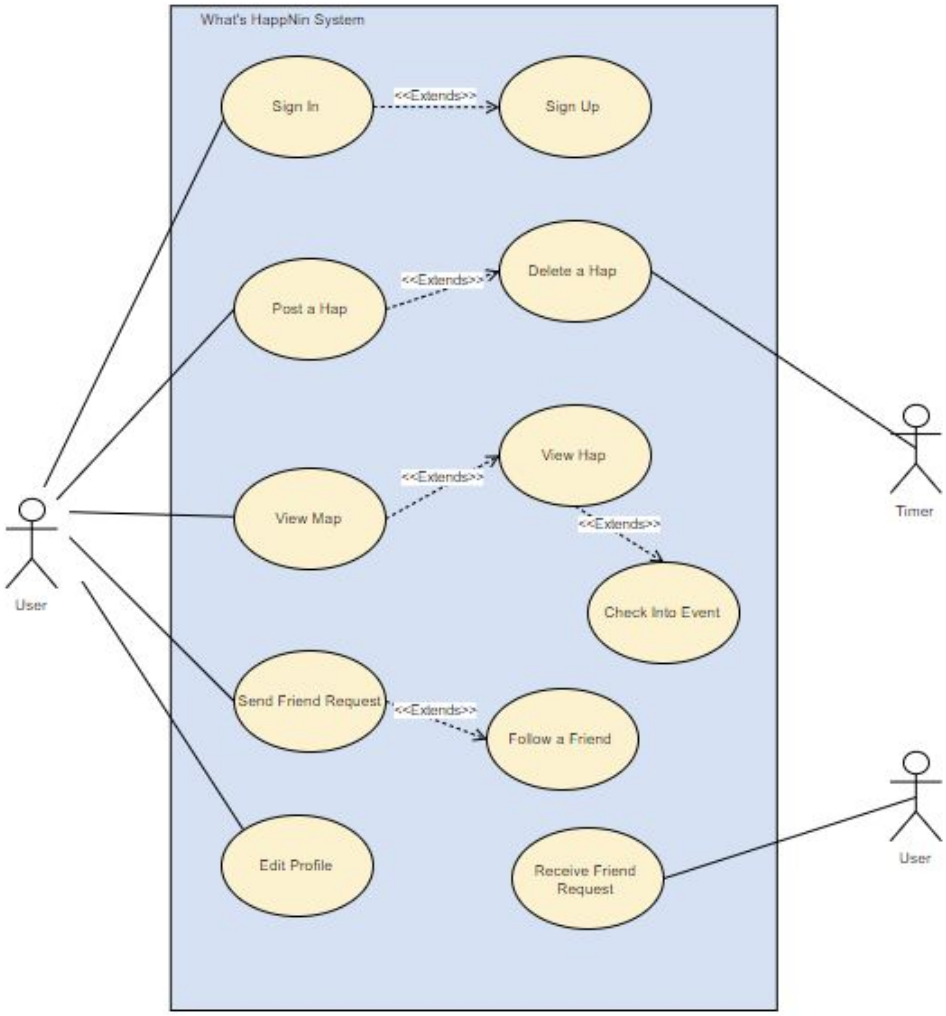
3.1.11. Edit profile

Description: Signed in User wants to Edit their profile.
User story(s): 4,19
Actors: User, Profile Database, Application.
Preconditions: User has an account and has already signed in.
User steps:

1. Starts up the application
2. Selects to view profile
3. Selects to edit
4. User can edit their current blurb or update their profile picture (can select to remove picture).

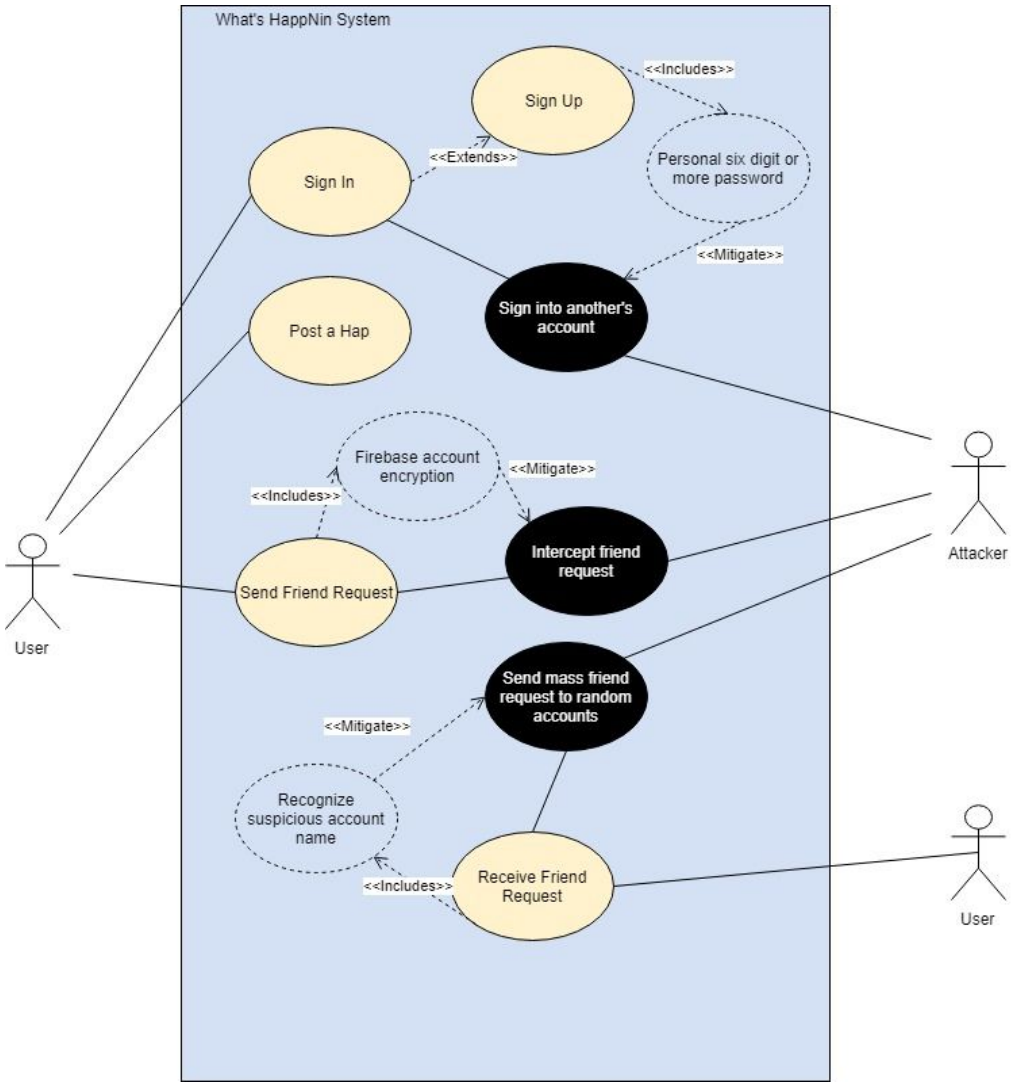
System steps: Pulls their current profile information from Profile Database, displays this to the user, once edited, the new information will be saved over old information in the Profile database.
Post-conditions: The User's new profile information will now display whenever their profile is viewed.
Exceptions handled: None.
Following Use Cases: None.

3.1.11.Use Case Diagram



3.1.12 Misuse case Diagrams

Diagram 1:

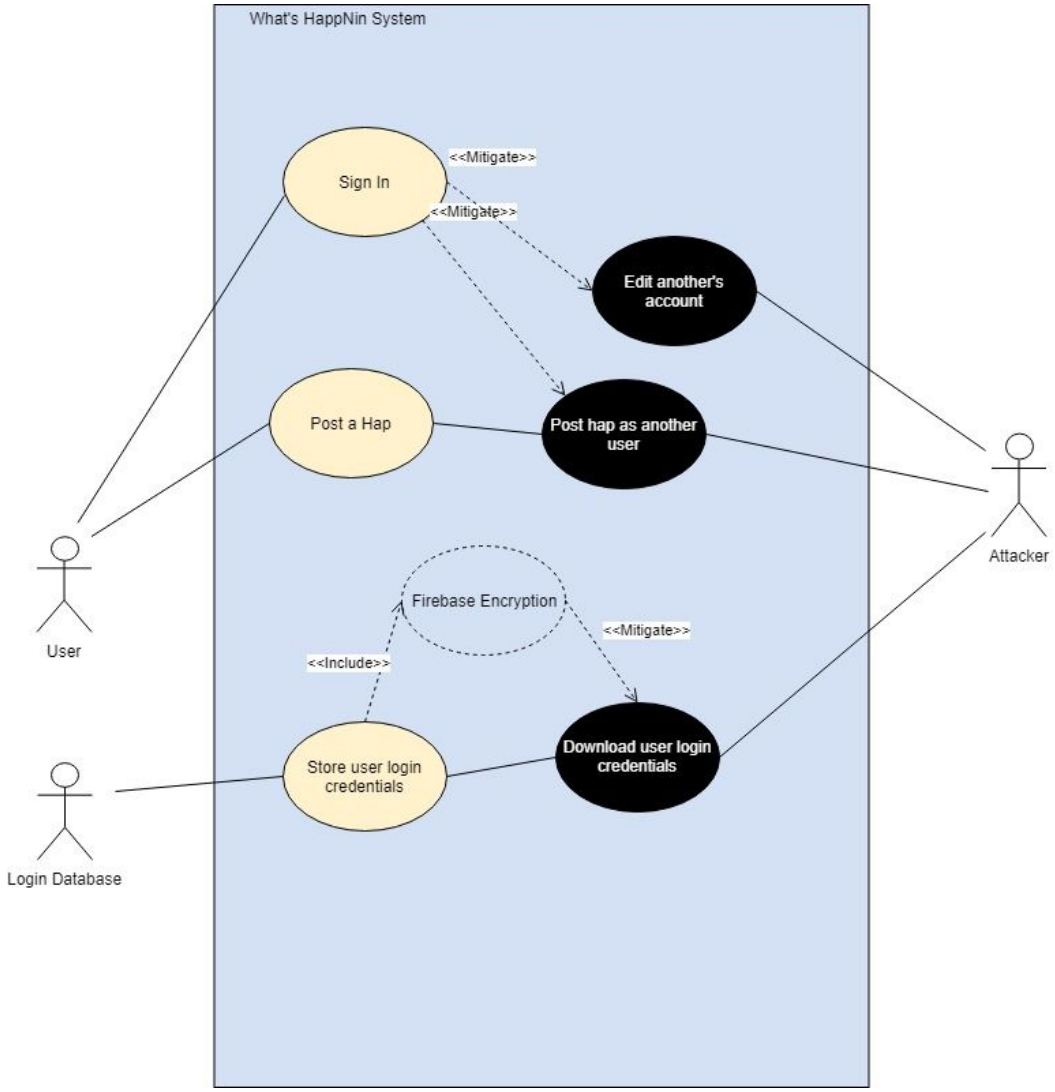


Misuse case 1: Sign into another's account
Mitigation technique: Require for the user to have a hard enough password of at least 6 digits during Sign-up process.

Misuse case 2: Intercept friend request
Mitigation technique: Use of Firebase encryption will ensure the request will make it to the correct recipient.

Misuse case 3: Send mass friend request to random accounts
Mitigation technique: User will be able to recognize suspicious friend requests and will not accept them.

Diagram 2:

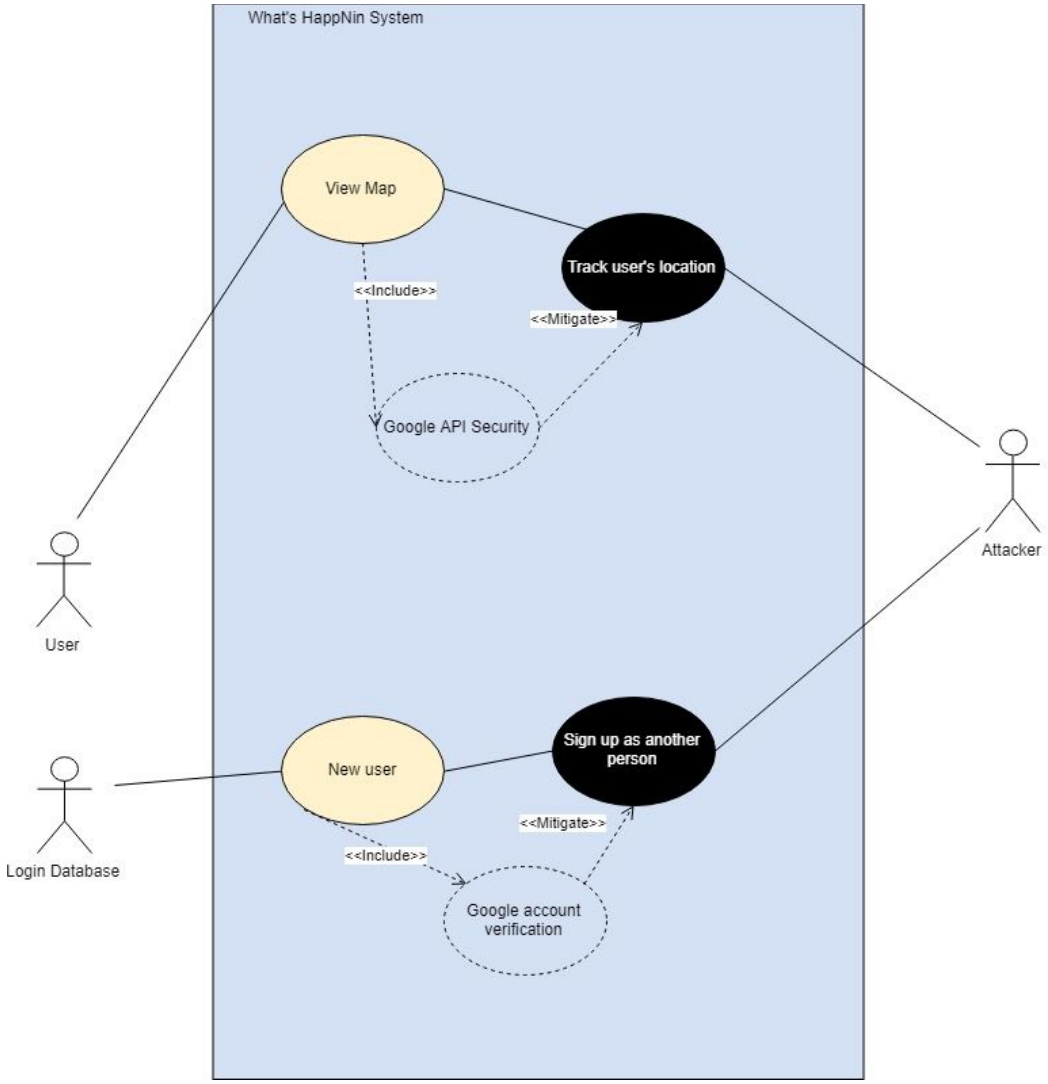


Misuse case 1: Edit another's account
Mitigation technique: Secure login will stop them from editing their account.

Misuse case 2: Post hap as another user
Mitigation technique: Secure login will stop the attacker from posting haps using another's account.

Misuse case 3: Download user login credentials
Mitigation technique: The database will store the user login information securely by using Firebase encryption.

Diagram 3:



Misuse case 1: Track user's location
Mitigation technique: When the application refreshes the location when accessing the map, it will use Google API which uses Google security protocols.

Misuse case 2: Sign up as another person
Mitigation technique: The sign up system will require the user to verify account creation using a Google account.

3.2 Functional Requirements

3.2.1. Sign Up/ Sign In

- **FR:** The system shall allow the user to create an account.
- **FR:** The system shall require a unique username.
- **FR:** The system shall require a six character password with at least one number.
- **FR:** The system shall allow the user to sign in with a username and password.
- **FR:** The system shall allow the user to remain signed in.
- **FR:** The system shall allow the user to sign out.

3.2.2. Profile

- **FR:** The system shall allow the user to upload a profile picture from their phone gallery.
- **FR:** The system shall create a Hap ranking based on the number of events they have attended.
- **FR:** The system shall display the Hap ranking on each user's profile.
- **FR:** The system shall allow the user to create a 100 character bio for their profile.
- **FR:** The system shall allow the user to see a list of their friends.
- **FR:** The system shall allow the user to select a friend off their friends list and follow them.
- **FR:** The system shall allow the user to edit their profile information.

3.2.3. Map

- **FR:** The system shall allow the user to view events on the map based on their location.
- **FR:** The system shall display the icon of the type of event on the map.
- **FR:** The system shall allow the user to select an event to event view information.

3.2.4. Selected Event

- **FR:** The system shall display the title of the event on the event feed.
- **FR:** The system shall display the description of the event on the event feed.
- **FR:** The system shall display the number of people at the event on the event feed.
- **FR:** The system shall display the media the users have posted to the event on the event feed.
- **FR:** The system shall allow the user to check in to the event when they are within the event's radius.

- **FR:** The system shall allow the creator of the event to edit the description.
- **FR:** The system shall allow the creator of the event to extend the event.
- **FR:** The system shall allow the creator of the event to delete/cancel an event.

- **FR:** The system shall allow the users who are checked into an event to post comments to the event feed.
- **FR:** The system shall allow the users who are checked into an event to post pictures to the event feed.
- **FR:** The system shall allow the users who are checked into an event to post videos to the event feed.

3.2.5. Create Event

- **FR:** The system shall allow the user to create an event based on their location.
- **FR:** The system shall allow the user to select an icon for the type of event.
- **FR:** The system shall allow the user to create an event description.
- **FR:** The system shall allow the user to set a time duration for a new event not to exceed 24 hours.

3.2.6. Menu

- **FR:** The system shall allow the user to create an event from the menu.
- **FR:** The system shall give the user the option to open the profile from the menu.
- **FR:** The system shall allow the user to open the map from the menu.
- **FR:** The system shall allow the user to add new friends from the menu.

3.2.7. Additional System Requirements

- **FR:** The system shall display the number of people checked in at an event.
- **FR:** The system shall decline a username when it is taken.
- **FR:** The system shall allow the user to send and receive friends requests.
- **FR:** The system shall allow the user to accept or decline friend requests.
- **FR:** The system shall allow the user to see the activity their friends are participating in.
- **FR:** The system shall alert the user when they are in an event radius so they can choose to check into the event.
- **FR:** The system shall allow the user to post media to the event live feed.
- **FR:** The system shall allow the user to send/receive messages to and from friends.
- **FR:** The system shall allow the user to follow their friends and be notified when that friends post to or join an event.
- **FR:** The system shall have a "do not disturb", so that the user won't be notified of anything.
- **FR:** The system shall require the user to input an email address to create an account.
- **FR:** The system shall authenticate the user using unique passwords.
- **FR:** The system shall get permission from the user before accessing their location.

3.3 Non-Functional Requirements

3.3.1. Performance Requirements

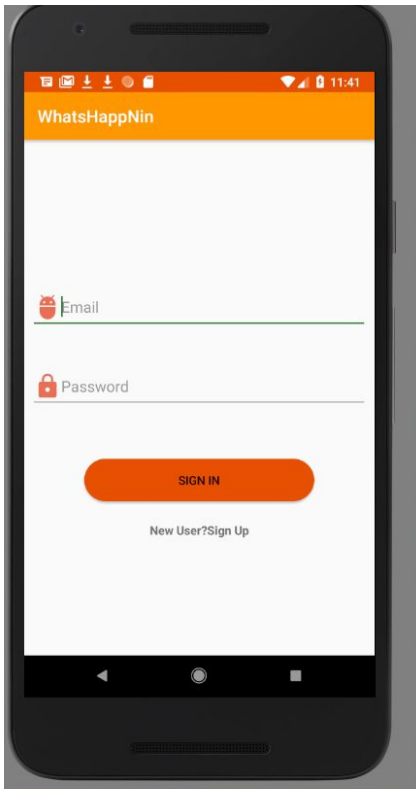
- **NFR 1.1:** The system response time must not exceed 3 seconds under the typical workload, 95% of the time.
- **NFR 1.2:** The system shall require only three steps to create a profile.
- **NFR 1.3:** The system shall allow the user to post in less than 15 seconds.
- **NFR 1.4:** The system map shall update everytime the map is opened.

3.3.2. Security Requirements

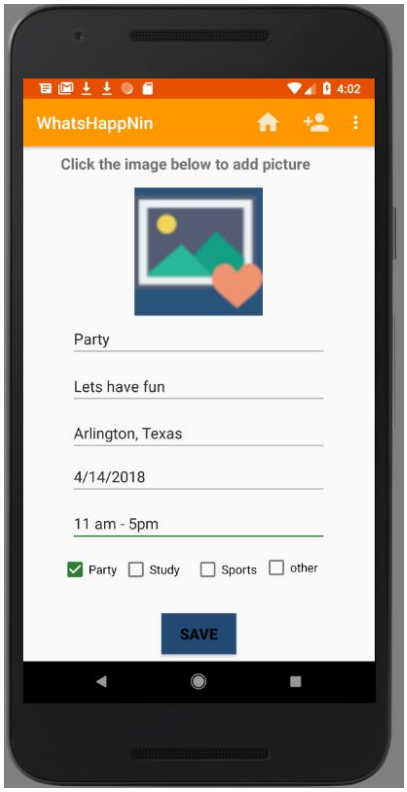
- **NFR 2.1:** The system shall encrypt the location data stored on the device.
- **NFR 2.2:** The system shall encrypt the account information stored on the device.
- **NFR 2.3:** The system shall verify the email address of a new user before creating an account for that user.

3.4 Interface Mockups

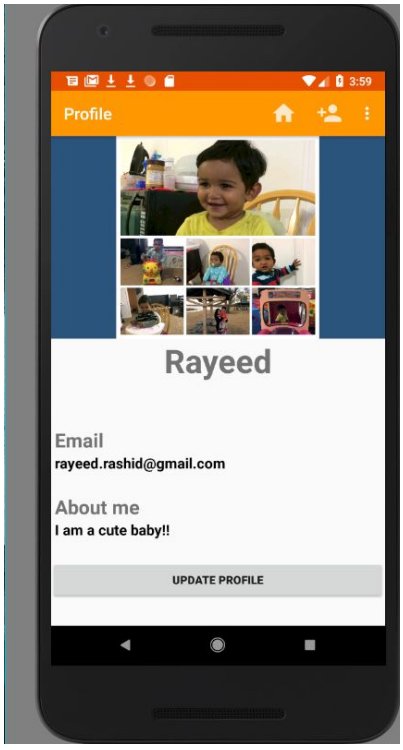
3.4.1. Sign-in



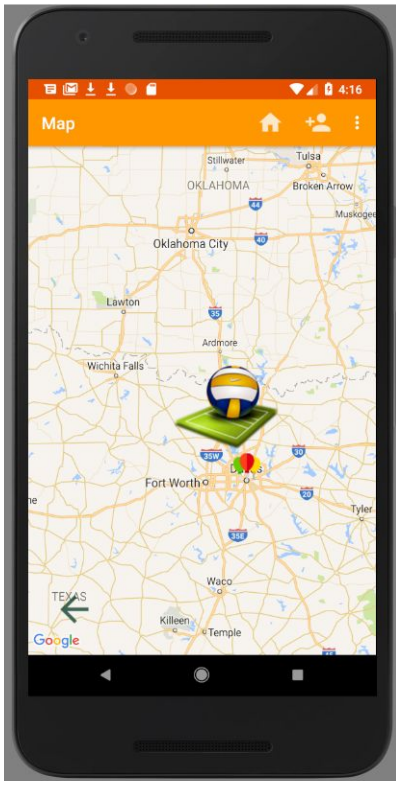
3.4.2. Create a Hap



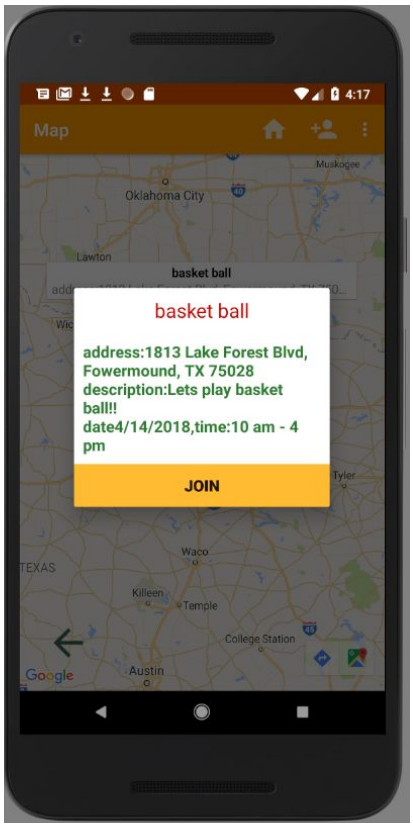
3.4.3. View Profile



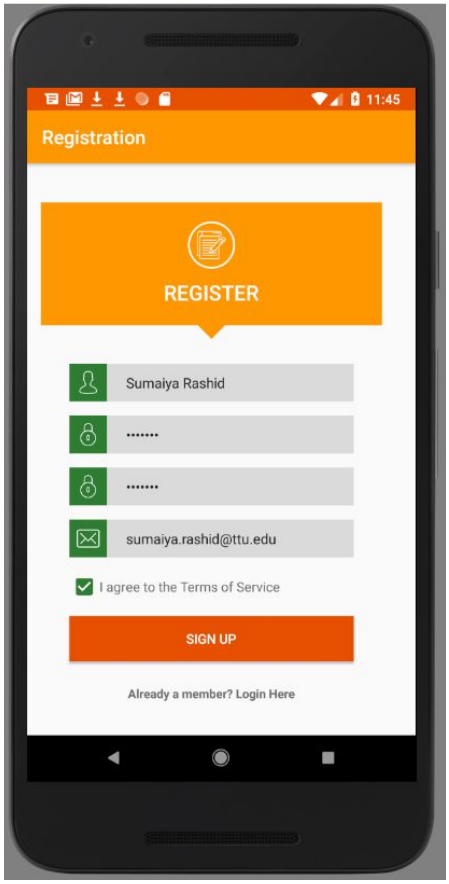
3.4.4. View Map


4:16
Map
Stillwater
Tulsa
OKLAHOMA
Broken Arrow
Muskogee
Oklahoma City
Lawton
Ardmore
Wichita Falls
Fort Worth
Tyler
Waco
TEXAS
Killeen
Temple
Google

3.4.5. View Hap


4:17
Map
Muskogee
Oklahoma City
Lawton
basket ball
basket ball
address:1813 Lake Forest Blvd, Fowermound, TX 75028
description:Lets play basket ball!!
date4/14/2018,time:10 am - 4 pm
JOIN
Tyler
Waco
TEXAS
Killeen
Temple
College Station
Google
Austin

3.4.6. Register



4. Focus Group Artifacts

4.1 Interview Script & Questionnaire

4.1.1. Interview Script

Welcome and thank you for being here today. The purpose of this gathering is to get your feedback about our new social app "What's HappNin".

Let me Introduce myself. I am _____ and I will be the moderator in today's discussion. Let me give you the consent form, and your names and information will be anonymous in our final report. We will be also video recording this focus group. The video recording will only be used to make sure our notes are correct, and will not be seen by anyone other than the group members and the professor. Now I will briefly explain what is our app about.

- General overview of the app?

- How the app will look like?
- What can we do with the app?
- How is this app different from other apps?

4.1.2. Interview Questionnaire

Q.1: Are you currently using any social app?

Q.1.a: If yes, do you ever use those apps to invite or join an event?

Q.2: What do you think about our app?

Q.2.a: Do you think this app will be useful for you?

Q.3: What do you suggest we should change or improve about this app?

Q.3.a: Any additional comment?

4.2 Focus Group Results

4.2.1. Focus Group A *(3 participants)*

Q1: All of the participants answered that they use at least one social media application. Two of the participant answered that they don't use social media apps to create/attend events. However, one of the participant specified that she uses Facebook, and she also answered that uses social apps to create events.

Q2: Two of the participants showed interest in our app, but they don't think this app will be useful to them. However, the participant who uses social app for events was undecided if this app would be any different than the app(s) she is using for social events. Her opinion of this app is that it might be useful to her.

Q3: The participants of Group A do not have any suggestions about how we should change or improve the app. One of the participants commented about how this app would be useful. According to the participant, this app would be useful for freshmen who just started college or people who recently moved to a new town and do not have any friends. They would become friends on the app with people they just met, and they would be able to meet with them again at events by using this app. This way the user could make more connections.

4.2.2. Focus Group B *(1 participant)*

Q1: The participant answered that he uses Facebook and Messenger. He said that he uses Facebook to invite people to orto join events.

Q2: The participant said that this app is similar to other apps. He said that he would use this app if it had unique feature.

Q3: The participant suggested that this app should be easy to understand and more user friendly.

4.2.3. Focus Group C *(1 participant)*

Q1: The participant answered that he uses Facebook and Snapchat. He said that he uses Facebook to invite people to or to join events.

Q2: The participant said that he would use this app.

Q3: The participant suggested that to add the ability to see all the people who are at the event, even if they are not your friend.

4.3 Focus Group Summary

According to our focus groups, most people use Facebook and some people also use similar apps like Snapchat and Messenger. Most of them said that they would use our app if it had unique feature. Some of them suggested that we add additional features such as the ability to view everyone at an event and not just your friends. We will take the participant's answers, suggestions, and comments into consideration, and make this app more user friendly and more unique than similar apps.

5. Appendix A: Consent Form

The following Focus Group Consent Form was presented to and signed by all focus group participants before conducting the focus group:

TEXAS TECH UNIVERSITY

Department of Computer Science

Focus Group Consent Form

Project Title: Whats HappNin

Investigators: Yavuz Kocoglu, Sumaiya Rashid, David Lucero.

- **I agree to participate in this focus group carried out by Yavuz Kocoglu, Sumaiya Rashid, David Lucero of the Texas Tech University, to aid the research of the Whats HappNin app.**
- **I understand the topics to be discussed in the focus group.**
- **I am fully aware that I will remain anonymous throughout the process of data collection and that I have the right to leave the focus group any time as I wish.**
- **I understand that the data collected will be stored securely and safely according to the Data Collection Act (1998).**
- **I am fully aware that I am not obliged to answer any questions that I do not want to.**
- **I agree to have the focus group meeting recorded.**
- **I am aware that I can make any reasonable changes to this consent form.**
- **I have read the consent form and understand the aims of the project.**

_____
Printed Name

_____ _____
Participants Signature **Date**

_____ _____
Researchers' Signature **Date**

Contact Information
Name of the Investigators: Sumaiya Rashid Yavuz Can Kocoglu David Lucero
Email of Investigators: sumaiya.rashid@ttu.edu yc.kocoglu@ttu.edu david.lucero@ttu.edu