# Basic User - Functional Requirements

## Use Case 1: User Login

### Introduction

* User should be redirected to the login page when he/she opens the app for the first time.
* Basic User should be given option to register as a new user.
* User will be auto logged in after the first login.

### Inputs

User should provide username and password in the respective fields and click on Login.

* **Username:** The username can consist of small case letters, upper case letters and numbers. It must have a minimum of 3 characters and a maximum length of 15 characters.
* **Password:** The password must consist of at least one small case, upper case, a number and a special character. The password should be minimum length of 8 and maximum length of 16. Password is encrypted before storing it in database.

### Processing

* If user clicks on Register button, user will be redirected to registration page.
* User should be validated by the respective authentication server based on the credentials provided.

### Outputs

If the login is successful, user should be redirected to “How to use app?” screen. “How to use the app” will be a short video/picture tutorial to help users recognize on how to use the app.

### Error Handling

If the user credentials provided are invalid an error should be thrown.

## Use Case 2: Registering User

### Introduction

Basic User should be redirected to the registration page when she clicks on “Create account” button in the login page.

### Inputs

Basic User should provide the following details to register as a new user

* Choose a Username

The username can consist of small case letters, upper case letters and numbers.

* E-mail

The email provided should be in the standard email format.

* Contact number

Should only be numbers with length 10.

* Choose Password

The password must consist of small case, upper case, numbers and special characters. The password should be minimum length of 8 and maximum length of 16.

* Re-enter password for confirmation purposes

The entry in this field should match with password entered in the above field.

* Profile Picture

The image will be resized to the predefined resolution.

* Birth date (translated to age in years)

Age of the user is calculated based on the date entered in this field.

* University

This is a dropdown. As of now, this app is available for only Northwest Missouri State University students.

* Greek affiliation if any (optional field)

### Processing

* Application checks the availability of the username.
* Applications checks if the passwords match.
* New user details are added to the database.
* App should send a confirmation SMS message to the phone number entered to make sure the correct phone number is linked up to the app.

### Outputs

If registration is successful, user should be redirected to “Home” screen.

### Error Handling

* If passwords do not match, an error will be thrown.
* If the username is already in use, user will be prompted to re-enter a different username.

## Use Case 3: Forgot Password

### Introduction

If the user forgets the password, she can reset her password by providing the user id.

### Inputs

User needs to provide “User ID” of her account and click on “Reset” button

### Processing

Based on the user id provided by the user, app fetches the corresponding email of the user from database. It is mandatory that every user should provide their email ID during registration.

### Outputs

An email is sent to the user with details to reset the password.

### Error Handling

If the user id provided by the user is invalid, app should show an error message.

## Use Case 4: How to use this app?

### Introduction

User should be redirected to “How to use this app?” screen, when user logins for the first time.

### Inputs

User will be provided with a slider that contains images/ Video that instructs user on how to use “PartyGuard” app.

### Processing

* Images and Video links are shown on the page in a slider.
* This screen is not shown if the user is not logging in for the first time.

### Outputs

User can view Images or Videos provided in this screen.

### Error Handling

If the Images and Videos are not loaded from the database, app should not crash.

## Use Case 5: Fraternities Screen

### Introduction

The List of fraternities that are available in the user’s location are enabled for user to select. Along next to the names should be a small image of the fraternity house provided by the Host user via the website.

### Inputs

Fraternities list in the user’s location are fetched from the database.

### Processing

If the fraternity has not paid for the usage of this app, the fraternity option is not enabled in this screen.

### Outputs

Fraternities list in the user’s location are fetched from the database. User can select a fraternity to report the issue.

### Error Handling

If the fraternity list is not loaded from the database, app should not crash.

## Use Case 6: Report Issue Screen

### Introduction

User can send an alert using this screen.

### Inputs

* User can select from a category of the issue from the list. The issues that are shown in this screen are as below
  + Accident
  + Feeling Unsafe
  + Fight
  + Other
* User can also select emergency button which would then ask the user if they were sure if they wanted to send the alert. If it is confirmed this feature allows the user to skip the location screen in order to send the alert faster. This is the only button that will trigger the University Police alert after 15 minutes.

### Processing

A timer is enabled to track if the issue is not resolved for more than 15 minutes, only if the emergency button is clicked.

### Outputs

* User can select an issue category to report it to the Fraternity. A confirmation message is shown on the screen triggering a push notification to Guard and Host users
* An emergency alert can be sent only by the EMERGENCY button to “Campus police” if the issue is not resolved by Fraternity within 15 minutes. A confirmation message is shown on the screen.

### Error Handling

If the fraternity list is not loaded from the database, app should not crash.

## Use Case 7: Report Location Screen

### Introduction

User can choose his/ her location in this screen so that Fraternity team can locate the person.

### Inputs

User can choose a floor (or) location within the building in this screen.

### Processing

* The location (or) floor list in this screen is populated based on the data provided by host user during registration of a Fraternity house.
* The Basic User then is sent to an Alert Summary page. They then have the option to cancel, add optional comments, and/or hit Send.
* A notification is sent to the fraternity member, after selecting this option.

### Outputs

The location selected in this screen will be included in the notification that is sent to the respective Fraternity members.

### Error Handling

If the location list is not loaded from the database, app should not crash.

## Use Case 8: Resolution Alert

### Introduction

Host user can push a notification to the basic user (user who submitted the issue) to confirm if the issue is resolved. A pop-up message is shown on the screen of the basic user to confirm the resolution of the issue.

### Inputs

User can either can choose “Accept” or “Decline” option from the pop-up.

### Processing

The option chosen by the basic user from the popup (Accept or Decline) is notified to the Host (or) Guard user.

### Outputs

**Accepted**: Issue is marked as resolved

**Declined**: The issue retains in “Un-resolved” state. Host user can re-notify the user once the issue is resolved.

### 3.8.5 Error Handling

App should not crash once the user selects the feedback.

## Use Case 9: Profile Dropdown Menu (or) Swipe Screen

### Introduction

This screen can be navigated by swiping from the left end of the screen (or) by clicking the dropdown menu () from the top left end of the screen.

### Inputs

User can select one of the below options to view (or) modify the existing settings

1. Profile
2. My Alerts
3. Location Change
4. Log Settings
5. About PartyGuard

### Processing

User can navigate to the respective settings screens by clicking the options provided in the list.

### Outputs

User can navigate to the respective settings screens by clicking the options provided in the list.

### Error Handling

If the options list is not loaded from the database, app should not crash.

# User case: Host User Login

## Use Case: Login Page

* The host user would be redirected to the login page when he/she opens the app and clicks on the Host User Login.
* User will be auto logged in after the first login.

### Inputs

* The user should provide Username and Password in the respective fields and click on the Login button.
* Username: The username can consist of small case, upper case letters and numbers. The username should consist of a minimum of 3 characters to 16 characters.
* Password: The password must consist of at least one small case, upper case, a number and a special character. The password should be minimum length of 8 characters and maximum length of 16 characters.

### Processing

* After the user enters the username and password and clicks on login button, he will be redirected to the PG Team Tab.
* User should be validated by the respective authentication server based on the credentials provided.

### Outputs

* If the login is successful, user will be redirected to PG Team Tab.

### Error Handling

* If the user credentials provided are invalid an error should be thrown.

## Use Case 2: PG Team

### Introduction:

* This screen will be the host landing screen, when the host user logs into the account he/she will be able to see this screen.
* This screen shows the guard members who are in that fraternity.
* When the host user clicks on a member, his/her profile will be displayed.
* The “green” circle on the image indicates the members who will respond to the alerts that were received.
* The members who are not in green in this screen will not receive any alerts.
* The user can select other tabs like alerts, history, and profile.

### Inputs:

* The user just need to click on a member to view his/her profile. Host user can switch on/off the option of receiving the notifications for this user. Rest of the profile changes can be done only from the website.
* As mentioned earlier, the “green” circled members are those who respond to the alerts. The assignment to the members will be done by the host user in the website or using app.

### Processing:

* When the user clicks on the members, their profile will be displayed.

### Outputs:

* The profiles of the guard members will be displayed when the host user clicks on the image.

### Error Handling:

* When this page is loaded, only the guard members of that fraternity should be displayed.

## Use Case 3: Alert Screen

### Introduction

Host user should be redirected to the alerts page when he/she clicks on “Alerts” tab.

### Inputs

List of alerts reported to the fraternity are shown in this screen. Host can select an alert from this screen.

### Processing

* The information regarding the alert, will be populated. The time elapsed after submitting the issue will be shown on the screen.
* If no one is handling that particular alert, “claim” button will be enabled. Any can claim the alert and confirm the claim.
* The name of the person who claimed the alert will be sent to the basic user. To all the other guard members “Claim” button in the screen will be replaced by the name of person who claimed the alert.
* After claiming the alert, any guard user who receives the alert can send the resolution confirmation request to the basic user by clicking on “Resolve” button.
* The basic user can accept (or) reject the request from the

### Output

The fraternity member will get the confirmation message once the basic user accepts the resolution request from the fraternity member.

### Error Handling

The app should not crash if the alert information is not loaded on to the screen.

## Use Case 4: History

### Introduction

The host user can be able to see issues that were resolved from the past few days when he clicks on the “History” tab.

### Inputs

A click on the “History” tab.

### Processing

When the basic user confirms that the issue was resolved, only then that issue will be displayed under “History” tab.

### Outputs

A list of resolved issues will be displayed with the name of the person and location.

### Error Handling

If there are no previously resolved issues, nothing will be displayed and the app shouldn’t crash.

## Use Case 5: Host/Guard Profile Screen:

### Introduction

Host/GuardUser should be re-directed to the profile page when he clicks on “Profile” tab in the Host/Guard landing screen.

### Inputs

Host/GuardUser could edit his profile by providing any of the following details.

* + First Name
  + Last Name
  + Email Address
  + Phone Number
  + DOB
  + PIN

### Processing

* Application directs to the profile screen.
* User can edit and change his (or) her details in the screen.
* New user details are updated to the database.

### Outputs

The updated details are shown on the profile screen of Host/Guard user.

### Error Handling

The application should not crash, if the page is loaded with new details.