

Appiness: Requirements Document

Ashlyn Cooper
Clemson University
Clemson, SC, USA
ancoope@clemson.edu

Charles Germany
Clemson University
Clemson, SC, USA
cgerman@clemson.edu

Evan Bondura
Clemson University
Clemson, SC, USA
ebondur@clemson.edu

Zachary Clay
Clemson University
Clemson, SC, USA
zclay@clemson.edu

Project Purpose

This project involves the prototyping and development of Appiness, a mobile application that focuses on the mental health of the user. Designed with college students and the stress of college life in mind, this mobile mental health application project has a defined goal of promoting more positive interactions between users and their phones. To achieve this goal, data and suggestions were taken directly from potential users on possible features of the application, ranging from remembering past positive accomplishments to daily positive quotes amongst other pitched features. From this, the vision of Appiness was fully realized and has led our team into developing requirements for the mobile application. Overall, through Appiness, this team seeks to provide self-dignifying reminders and methods of improving and maintaining mental health to its users as they deal with the stress and anxiety of higher education.

User Groups

Appiness is being designed with a limited and specific user group in mind, however this specific user group has wider variability within it. As stated in the project purpose, Appiness is being designed to help relieve the stress of college students or other young adults. While this user group is constrained within a tight age frame, mainly 18-25 year olds, the users that make up this group may come from a wide variety of backgrounds and have different experiences with mental health. Through our questionnaires and user interviews, we were able to gain valuable information with regards to the types of features that may be desired by potential users as well as learn more about exactly who our potential users are. Generally speaking, the potential users of this app experience varying levels of stress throughout their daily lives and are adept at using smartphone applications. From these generalizations, we have derived two personas from our potential user group. Each of these personas represent a hypothetical archetype of an actual user of Appiness. Both of these personas may be using the app to achieve similar goals, however each may have a slightly different background and therefore different needs and goals when using the application. The personas and their descriptions can be found in Figures 1 and 2 below.

Personas

Below are two images describing the personas derived from our user group. The first persona, Charlotte in Figure 1, is at the tail end of our target user demographic at 24 years old. She also represents a distinct user in that she is out of college and also does not suffer from any chronic or diagnosed mental health issues. The second persona, Adam in Figure 2, is at the younger end of our target user demographic. Adam represents a potential user that is currently attending college and also consistently deals with his own mental health issues on top of daily stressors experienced by most if not every person.

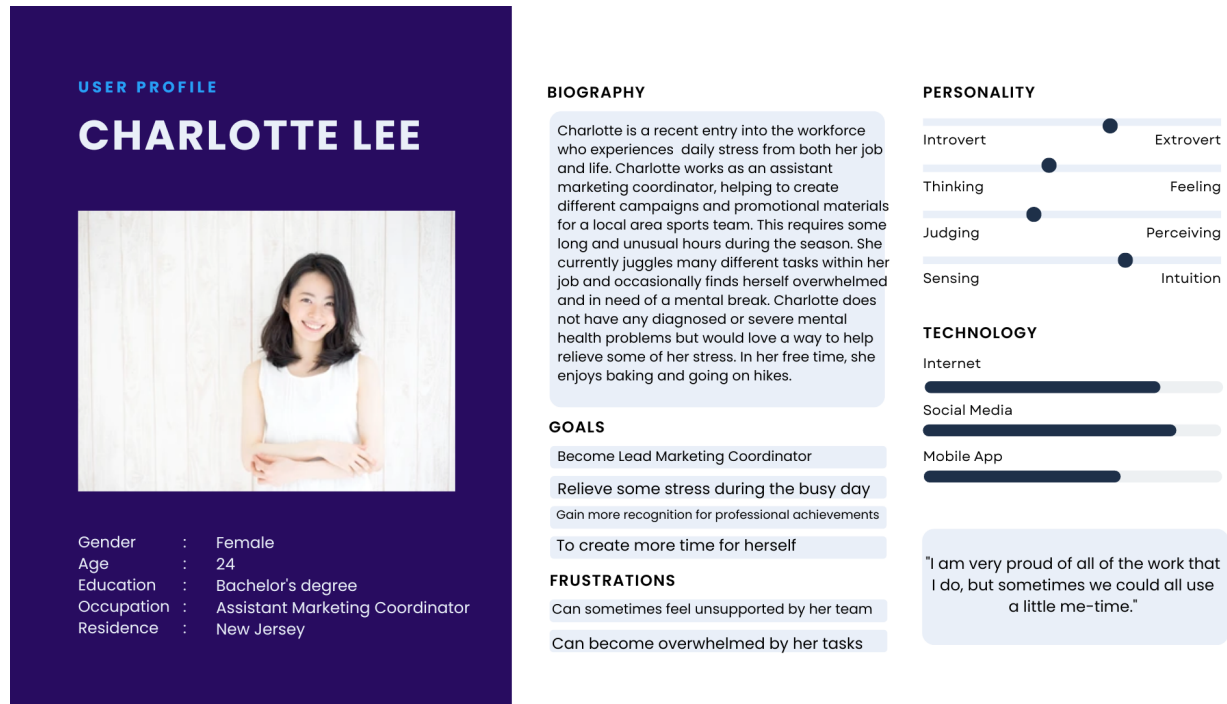


Figure 1

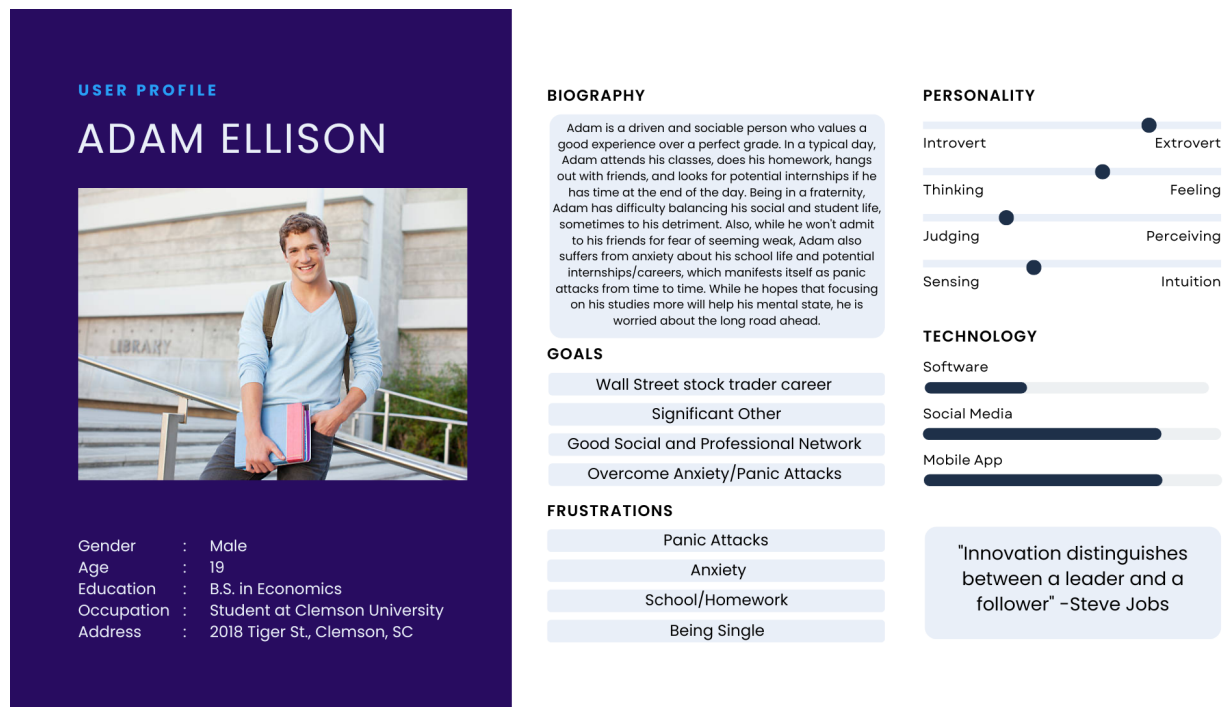


Figure 2

Volere Shells

Requirement 1	Remembering Past Accomplishments
Type	Functional Requirement
Description	The user should have the ability to view their past accomplishments anytime on the app.
Rationale	This feature will give the user the ability to look back on all that they have accomplished. This will positively contribute to their overall happiness if they are able to remember the good work that they have done.
Source	Users
Use Case	<p>Preconditions: The user is already logged into the application. The user is on the home screen.</p> <p>Normal Course:</p> <ol style="list-style-type: none">1. The application displays the home screen upon launching.2. The user clicks a button allowing them to view their past accomplishments.3. The user can then scroll down a timeline that shows all of the past accomplishments that they have entered into the app.4. The user can click a button if they want to add an accomplishment.5. The user inputs their accomplishment date and description, then the user hits save.6. The user then goes back to the accomplishments screen and can return to the home page with a button.
Objective Test	Once a user is logged in, they should be able to access their past accomplishments from a button on the home screen. The past accomplishments screen will then appear, allowing the user to scroll through all of their accomplishments.

Requirement 2	Daily Message
Type	Functional Requirement
Description	Each day, the app will provide a daily message that contains a motivational quote or stress relief exercises.
Rationale	When the user reads the daily message, they will be more motivated throughout their day. It is also a good reminder for the user to take some time to relieve stress and put their mind on good things.
Source	Users

Use Case	<p>Preconditions: The user is already logged into the application. The user is on the home screen.</p> <p>Normal Course:</p> <ol style="list-style-type: none"> 1. The application displays the home screen upon launching. 2. The home screen contains a daily message intended for the user to read. <p>Alternate Course:</p> <ol style="list-style-type: none"> 1. The user receives a notification containing the daily message. <ol style="list-style-type: none"> 1.1 The user clicks the notification, which launches the app and displays the home page.
Objective Test	Every day at a specified time, the user's mobile phone will receive a notification with a daily message. When the user is logged in, the daily message will be displayed on the home screen.

Requirement 3	Set Number of Checkins
Type	Functional Requirement
Description	The user is able to specify the number of times the application notifies them daily.
Rationale	We don't want the application to spam the users mobile phone with notifications, so the user will be able to specify how many times a day they wish to be notified. Also, through user interviews we gathered that different users may want the app to notify them a different number of times.
Source	Users
Use Case	<p>Preconditions: The user is already logged into the application. The user has opened the settings menu.</p> <p>Normal Course:</p> <ol style="list-style-type: none"> 7. The user clicks on the "Notification Settings" option from the settings menu. 8. The app opens a screen displaying the current frequency of notifications. 9. The user clicks a button allowing them to modify the frequency. 10. An input field opens and the user is able to input the number of times a day they would like to be notified by the app. 11. The user then clicks a button to save the changes. 12. The application asks for confirmation of the changes and then, when confirmed, redirects the user back to the settings page. <p>Alternate Courses:</p>

	<p>4. The user enters an invalid number of notifications per day</p> <p>4.1 The application displays an error message and prompts the user to enter a valid number.</p> <p>5. The user decides not to make changes to their notification frequency</p> <p>5.1 They select the "Cancel" button to return to the notification settings screen without making any changes.</p>
Objective Test	When the user is logged in, they will be able to set the number of times they wish to be notified from the login screen. Once set, the desired number of notifications should be sent to their phone daily.

Requirement 4	Customizable User Interface
Type	UI requirement/Functional requirement
Description	The user should be able to modify their settings with in the app to change its appearance (color scheme, light/dark mode, etc)
Rationale	We want our app to be as personalizable as possible to allow users to build a stronger connection with the app and thereby facilitate more positive interactions between the app and its user base.
Source	Users
Use Case	<p>Preconditions: The user has logged into the application. The user has navigated to the settings menu.</p> <p>Normal Course:</p> <ol style="list-style-type: none"> 3. The user selects the "Appearance Settings" option from the settings menu. 4. The application presents the user with a screen displaying the current appearance settings, such as color scheme and light/dark mode options. 5. The user chooses the "Change Appearance" setting option. 6. The user then sees a menu displaying all the appearance changes they are able to make. 7. Next the user will select the changes they would like to make. 8. The application shows the user what the application will look like with their changes and then asks for confirmation and then returns the user to their settings page with the updated appearance enabled. <p>Alternate Course:</p> <ol style="list-style-type: none"> 6. The user decides not to change the appearance settings.

	6.1 The user selects the "Cancel" button and returns to the settings menu without making any changes.
Objective Test	Once a user is logged in they will be able to change the settings for the color scheme. These changes will be saved and remain as the way the user had changed it each time they log in.

Requirement 5	Daily Advice/Tips
Type	Functional Requirement
Description	When any user opens the application they will be able to see an advice column or mental health tips which will change on a daily basis.
Rationale	As the goal of Appiness is to improve the mental wellness of its users, the advice column or general mental health tips will help users better their mental health on a daily basis by giving them things to do or words of advice to increase their happiness.
Source	Developers
Use Case	<p>Preconditions: The user has launched the application.</p> <p>Main Course:</p> <ol style="list-style-type: none"> 1. The application displays the home screen upon launching. 2. The user is presented with a daily advice column or mental health tip on the home screen. 3. The user can read the advice column or mental health tip. 4. The application updates the advice column or mental health tip daily. <p>Alternate Course:</p> <ol style="list-style-type: none"> 1. There is no internet connection. <ol style="list-style-type: none"> 1.1 The application will display a message indicating that it is unable to update the daily advice column or mental health tips due to a lack of internet access.
Objective Test	As soon as the app opens for any user, the advice column and/or general mental health tips will be displayed.

Requirement 6	Mood Tracking
Type	Functional Requirement

Description	The user should be able to input their current mood at any point throughout their day.
Rationale	By doing this the app will be able to track the times when the user most often feels happy or sad throughout the day. This will allow the app to determine when notifications or daily messages should be sent to the user.
Source	Users
Use Case	<p>Preconditions: The user has launched the application and is logged in.</p> <p>Main Course:</p> <ol style="list-style-type: none"> 1. On the main screen the user is able to view some sort of mood scale or list of emotions that they can choose from. 2. The user selects the option that best represents their current mood. 3. The application records the user's mood for that time and day and stores it in the user's profile. 4. The user can view their mood history within the app. <p>Alternate Flows:</p> <ol style="list-style-type: none"> 2. The user chooses not to input their mood at this time. <ol style="list-style-type: none"> 2.1 This is a completely optional section of the application, so another course here would be the user choosing not to input their mood. 3. A connection error occurs. <ol style="list-style-type: none"> 3.1 This would lead to the application failing to receive the user's mood for the day. The app would then display an error message and ask the user to try again later.
Objective Test	The user can input their given mood within the app, and after some time, the app will begin to determine when it should send that particular user uplifting messages. A log of the user's previous moods can be seen.

Teamwork Section

To begin this assignment, we as a team first met together for an hour to review the paper requirements and to ensure that we had a clear plan of accomplishing this assignment. Within this collaborative brainstorming session, we considered our previous user research into the potential features of the application and used it to collectively develop six basic ideas for requirements. After that, we then divided the necessary sections of this paper equally based on our individual comfort and capabilities. The contributions are as follows:

- Evan completed the "Project Purpose" introduction, the male user persona, and the "Teamwork Section" itself.
- Ashlyn completed the "User Groups" and female user persona.
- Zach developed and completed three Volere Shells and objective tests for our requirements
- Charlie developed and completed the other remaining three Volere Shells and objective tests.

Once each section was completed, other teammates would proofread each other's work and suggest edits and improvements to increase flow and clarity of the overall paper. Lastly, when the whole paper was completed, each member reviewed it and gave it their stamp of approval for submission. Overall, it was a thorough and collaborative process by all team members to ensure the best possible version of this paper