1. Executive Summary

The Wander Watch for this Senior Design project is a watch that communicates with a hub and Android app in order to help keep track of the wearer of the watch. The watch communicates with the hub to show that the watch is within the designated area. If the bearer of the watch leaves the area, the hub communicates with the Android app to alert the user. The user can then check the app to see the watch-wearer’s location via GPS. Using this information, they can contact the right people in order to return the wearer of the watch to their home. This project was original meant to be used for older patients with dementia, who may wander from their homes. However, it is hoped that the watch can be used to also keep track of children, and expanded upon for other uses. In the following pages, the Android application will be the main focus. It is also hoped that this project will provide an affordable option to keeping track of loved ones.

The watch is designed to be easy to wear, unlike previous products which may have uncomfortable wearables, or small devices that may be easily lost. It will be able to display the current time and date, so that it can also act as a functional watch. The hub is designed to be able to track the watch within a reasonable range. It uses the best technologies that will allow it to track the watch and send alerts efficiently, while also keeping down its cost of production.

One of the major goals of the project was to include an application with user-friendly design. With mobile applications becoming more popular in the modern era, it was determined that having an app exclusive for this project would help with having to interact with the hardware of the project, namely the watch and the hub. Instead of having yet another device for keeping track of the watch, this project has an Android application that is easy to use. The main menus of this app are the map, alerts, and settings. The app will also allow users to log in with their own account, in order to keep track of their information. The simple menus and features are meant to streamline the project of checking on the watch-wearer and their location. This will be explained in more detail later on.

Besides the design of the app, this report will also explain how the watch, app and hub will function by themselves and with each other. This will also show how each aspect of this project will send the necessary information to each other. Such information includes the watch’s location, whether or not the watch is within range of the hub, and the content of the alerts that the application will receive from the hub.

In addition to the design and functionality of the three parts of this project, there will also be a brief view of the possible future expansion of the Wander Watch. This includes extra features for the watch, hub, and application. There will also be a discussion on improved functionality that would allow the hub and application to handle tracking more than one watch at a time. It is hoped that this project will be successful outside of the scope of the course, and that it helps a lot of people assist their loved one.

3. Existing Projects and Products

Over the course of the project team’s research, it has been found that there are a lot of existing products and projects that are similar to the idea for the Wander Watch. There are a lot of similarities, but also some major differences that will hopefully show that this project is still worthwhile as a Senior Design project. The following few sections go into detail on some of the other products that were found during research for the project.

One of the first articles that was found was one that listed “10 Lifesaving Location Devices for Dementia Patients” [10]. As can be deduced from the title, the article talks about 10 different devices used to keep track of dementia patients and ensure that they do not get lost. They each have their own features and differences compared to this project. Many of these features were taken into consideration, and the overall project was designed in a way to address both the benefits and pitfalls of some of these previous products.

The first is a program called Project Lifesaver. A person enrolls in the program and is given a transmitter. Once the person gets lost, their caregiver can contact their local Project Lifesaver agency, who will then track down the person via their transmitter. To enroll in the program, an initial $95 fee must be paid, in addition to paying $25 monthly [11]. Unlike what is planned for the Wander Watch, it seems that caregivers who enroll their patients into the Project Lifesaver program cannot actually track the patient themselves. In addition, they must fill out forms for enrollment, which could be a long process. The Wander Watch will instead allow people to purchase the watch and hub, set it up, create an account on the app, and track the watch themselves with no need of a middleman or form processing. The design of the watch itself is also meant to be comfortable to wear. For Project Lifesaver, the patient must wear their transmitter around their ankle. This can be uncomfortable for most people.

The second is called Mindme [12]. This is actually two devices made for dementia and elderly patients in mind. One device is called Mindme Alarm, which is used to call emergency services when the user has fallen or gotten hurt. The other device is the Mindme Locate, which uses GPS to locate the user if they have wandered away from where they are supposed to be. A caretaker can even set up a pre-set location that the user should not move out of. The major downside to this product is that it is only available in the European Union. Obviously, a product that is only available in a particular region will limit how many people have access to it. Plus the Mindme devices both come with a rather large initial fee, around £85 or $90, and monthly subscriptions. This makes them an unreasonable option for those who cannot afford the price and subscription. It is also a rather small device that needs to be carried around by the person who is being monitored. Since it cannot be worn, it is possible for the person to lose the device. This can be a risk for people who may wander away from home, or who get lost often.

The third product on the list is the GPS SmartSole [13]. This is a wearable tracking device in the form of a sole that is worn inside a shoe. So the device is placed inside the shoe of the person that needs to be tracked. Their caretaker can keep track of their location using an app. If they become lost, the caretaker can contact emergency services to safely recover them. The SmartSole even allows the user to set up a Geo-Fence for the person wearing the sole. This is a pre-set location that the person needs to stay within, similar to the one for the Mindme device. When the person with the sole walks out of the Geo-Fence, an alert will be sent to the caretaker, either through the app or by text messages. From there, they can alert emergency services and get the person back home. This is similar to how the project team is planning to handle alerts and messages for the Wander Watch via the application. However, the device itself costs $299, while also having a one-time activation fee of $35 dollars, plus a $24.95 per month subscription. This is incredibly expensive compared to most of the other products and projects that have been researched, and is most certainly unaffordable for the average consumer.

The same company is also planning to sell a Bluetooth version of the SmartSole [14]. This version has a battery life of about a year, which is close to what the project team plans to have for the Wander Watch device. It also comes with a “CUBE gateway”, which is described as an antenna device. This device allows the caretaker to set up a perimeter much like the Geo-Fence. It is plugged into a power outlet, and then the caretaker can set up the perimeter. They can also track the sole using an app, much like the GPS version. Though this Bluetooth version is also much cheaper than the GPS version, it also has a device price, activation fee, and a monthly subscription. Also, due to using Bluetooth, it is meant to only be used indoors and within a limited area, as it is incapable of tracking the sole over a wide range. This is much the same reason why the project team ultimately decided not to use Bluetooth for the Wander Watch. It is much more limited compared to GPS, and can make it difficult to track a person’s location past a pre-set location.

The fifth device is called the Safe Link tracker [15]. It is a device that must be carried by the person that needs to be tracked. According to the company’s website, it can act as a simple cellphone, and it also comes with an SOS button for emergencies. It sends its current location periodically to the Safe Link servers. The device is associated with a particular account, similar to what is planned for the Wander Watch. Then, caretakers can login to the Safe Link website to check the person’s location. If they feel that they have gotten lost or wandered away, the caretaker can then contact emergency services and make sure that the person is safe. However, the device seems to be rather small. In fact, it must be carried by the person being tracked. This can be a problem since it could be lost or misplaced. The device is priced at $169.99 for the device, and $18.97 for the subscription. While this is not as expensive as the GPS SmartSole, it is still probably too much for most people to afford.

The sixth device is called the PocketFinder [16]. This device can be used for tracking elderly people, though the website also has versions of the tracker for children, cars, and pets. The device comes with a its own app, through which caretakers can keep track of the user’s position via frequent location updates. They can also be sent updates via notifications on their phone, text messages, and e-mail messages. An interesting feature of the PocketFinder is that it will only send updates about its locations when it is in motion. This is meant to conserve battery power. Unfortunately, this device is small in size. In fact, it appears to be the smallest device that has been found during research. for this project. Though the PocketFinder website mentions that it can be put inside a pocket or strapped to a belt, it is still possible for the device to be loose and get left behind by the person who needs it.

The seventh device on the list is known as the Revolutionary Tracker [17]. On the website it is stated that this device is not only a GPS tracker, but also functions as a wearable watch and cellphone. This device allows for multiple people to be tracked by a single caretaker, by either their smartphone or computer. Similar to the Safe Link tracker, the Revolutionary Tracker also has an SOS button for emergencies. The watch itself is $199.00, and has a $25 per month subscription. Though this device is a wearable tracking device like the Wander Watch, it is rather expensive by itself. In addition to this, the Revolutionary Tracker does not seem to be available yet, as the website mentions “(Shipping soon)”. This could be because the product is still being developed, and is only taking pre-orders of the device for now.

The eighth device on the list is actually a program known as the Comfort Zone Check-In [18]. This program offers its own device that can be used to track a person, but it also offers the option to track the person’s own cellphone instead. Whether the caretaker chooses to use the separate device or a cellphone, they can login to a web-based software to track a person’s location. However, the cellphone can only be tracked if it is on the Sprint network. Like with previous devices, having to carry a small device around runs the risk of losing the device. Even if someone were to chose the cellphone option instead, then that would mean make sure the cellphone is being carried by the person being tracked. Much like the other products discussed so far, the Comfort Zone Check-In is not a cheap option. The device is about $100 and its subscription is $14.99 per month. If a caretaker wants to use a cellphone that the person already owns, then they would only have to pay for a $9.99 monthly subscription. The cellphone option seems like a very good idea. There is no need to buy a separate device, and the subscription is cheaper than most of the other devices listed in this section. However, the fact that the cellphone must be running on the Sprint network means that only people who already own a Sprint phone will have the chance to use the Comfort Zone Check-In program.

The ninth device is called the GPS Tracking Locator Watch, and is offered by a company called Bluewater Security [19]. The watch has the same size and appearance as a normal wristwatch, and has a battery life of about 30 days. The watch itself has a panic button for emergencies, and comes with a receiver. If the person with the watch gets a certain distance away from the receiver, it will set off an alarm to show that the person has wandered off. The watch also allows for messaging, so that the caretaker can communicate with the person while trying to track them down. The messages are only accessible on the receiver, and it does not have an option to check the messages via a phone or website. The watch and receiver together are $599.99, which is the most expensive price that has been found during research. The GPS subscription for the device is around $35.00 per month. So not only is the device itself the most expensive device on the list, the subscription is also the most expensive of all the subscriptions that have been seen so far. Some people would not be able to afford the watch and receiver, let alone a $35 per month subscription.

The final item discussed on the list does not involve any technological devices being worn or carried for tracking. It is the MedicAlert Safely Home program [20]. A person is registered into the program and given an ID bracelet with their information. If they get lost, their caretaker can call the police and inform them that the person has been registered with the MedicAlert program. From there, the police can contact MedicAlert directly and get information about “possible locations” that the person could have gone to. This implies that the company does not use GPS to track anyone, nor is there a tracker in the ID bracelet. This sounds like it could make it very difficult to actually find the person. In addition, this program only seems to be available in Canada, limiting its use to people within that country. Though the program only has a $60 per year subscription, it does not really differ much from traditional emergency services who do not use advanced technologies to track and recover lost people. Indeed, seeing this program has reaffirmed the team’s belief that using a GPS device like the Wander Watch would make tracking far easier for everyone involved.

Another product that was found during the research for the project was the Keruve Watch [21]. This is a watch that is tracked by a separate device known as the Keruve Receiver. This device can communicate with the watch via a mobile phone network, and track its location via GPS. However, both the Keruve Watch and the receiver have poor battery life compared to what is planned for the Wander Watch. Additionally, the website indicates that people must purchase the watch and receiver. It does not state how much the watch and receiver cost, but it can be assumed that the price is similar to the other products discussed in this section.

An interesting project that was discovered is the F\*watch [22]. This is a free source project for designing a watch with GPS tracking capabilities. It is not sold as its own product. Instead, everything that someone would need to build the watch themselves is available for free on the F\*watch website. This includes buying parts and building some of them with a 3D printer. The watch does not allow it to be track via other devices. Still, a lot of the hardware and software used in this project will be used as a basis for the Wander Watch project.

Overall, it is the hope of this Senior Design project that the Wander Watch will be a viable alternative to these previous products. The aim is to make it cheap enough that most people will be able to buy it easily, and be able to set up and use the watch without a needed for a middleman or additional services. Ideally, the hub itself will act as a middleman of sorts for the watch and mobile app. Besides this, most of the products that were researched have a device that the person must carry with them. These devices tend to be small so that they can be carried easily. However, at the same time this makes them easy to lose, since there is no easy way of attaching them to the person that needs to be tracked. This is a big reason as to why it was decided to make this project a watch. It will be comfortable to wear while also making sure that it cannot be lost by the user. It will also be functional, since it will be able to display the current time and date, like a regular watch. The project team have been taking into account all the positive features of these products, but also their downsides. It will be ensured that the Wander Watch will be as effective and useful, yet still affordable, as possible.

3.1 Benchmark Table

To conclude this section, the following table shows the major features of each previous product in comparison to the benchmark that has been planned for the Wander Watch. This is to show a visually convenient comparison between these products and what the project team plan to achieve with this project. This also gives a good way to track what the other products have done, and to check and see how well the project is meeting these benchmarks.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Comparison Benchmarks Table** | | | | | | | |
| Name | Range | Tracking | Availability | Device fee | Monthly fee | Includes a hub? | User can track the patient? |
| Wander Watch | 10-15 ft | SMS and GPS | Worldwide | $100 or less | None | Yes | Yes, on app |
| Project Lifesaver | 10-50 ft | Radio tracking | U.S. | $95 | $25 | No | No |
| Mindme | About 32 ft | GPS and SIM | European Union | Around $90 | Around $18.72 | No | Yes, on website |
| GPS SmartSole | 10-50 ft | GPS | U.S. | $299 + $35 | $24.95 | No | Yes, on app |
| Bluetooth SmartSole | Indoor only | Bluetooth | U.S. | $49 + | N/A | Yes | Yes, on app |
| Safe Link Tracker | 10-50 ft | GPS | N/A | $169 | $18.97 | No | Yes, on website |
| PocketFinder | 10-50 ft | GPS | U.S. | $129 | N/A | No | Yes, site and app |
| Revolutionary Tracker | 10-50 ft | GPS | U.S. | $199 | $25 | No | Yes, site and app |
| Comfort Zone Check-In | 10-50 ft | GPS and cellular | U.S. | $99.99 | $14.99 / $9.99 | No | Yes, on website |
| Locator Watch | 10-50 ft | GPS and GSM | U.S. | $599 | $35 | Yes | No |
| MedicAlert Safely Home | N/A | N/A | Canada | None | $60 per year | No | No |
| Keruve Watch | 10-50 ft | GPS | U.S. | N/A | N/A | No | Yes, on receiver |
| F\*watch | 10-50 ft | GPS | Worldwide | None | None | No | No |

Table ?.? - The comparison benchmarks table used for this project.

5. Android Application

Though there are many different kinds of mobile applications, it was decided that creating an Android application was the best option for this project. There are many factors that led to this decision. These include the fact that it is cheaper overall to develop Android applications, the majority of these apps are made using Java, Android offers a lot of tools and features that are helpful for the project, most of the team members already have experience with Android app development, and there are already plenty of tutorials and guides for working with Android that will help with development.

5.1 Android and Java

The first factor is that it is ultimately cheaper to develop an Android app than it is to develop an app for other mobile devices. One of the competitors against Android are the mobile devices developed by Apple. For both Android and Apple, it is free to develop applications with either of their SDKs and code. Apple uses Xcode to develop their apps. However, to distribute these applications through the App Store, a yearly membership is required [1]. The cheapest membership option is $99, which can be a very significant chunk of the project’s budget. In comparison, Android requires only a one time $25 fee to distribute an application on the Google Play store [2]. Additionally, developing an Android application will ensure that it is accessible to a larger number of people. As of the second quarter of 2015, Android smartphones make up 82.8% of the market share of smartphones. This means that Android smartphones are the most widely available phone on the market, and thus it is the one that people are most likely to buy and own. In comparison, iOS smartphones only took up 13.9% of the market during the same time period [3]. Since it is less costly in the long run, and Android smartphones are more widely available, this was the first major factor in deciding to create an Android app for the project.

Overall, the project team is currently not concerned with having the Wander Watch app be available on devices besides Android. While having compatibility across multiple devices would be a great thing to have, it is also very expensive. As mentioned above, there are fees that need to be paid before a developer is allowed to distribute their apps on a certain device. Larger companies with bigger budgets are able to release their applications across multiple devices. Examples of this are video games companies who can release their games on multiple consoles. They have enough money available to pay for both development of the game itself, and for the fees and licenses needed to publish the game on different consoles. As a college project, it is not expected that the Wander Watch will have enough of a budget to justify the cost of developing its application for more than just Android. Besides the cost, there would also need to be a lot of time spent coding the application for multiple devices. Earlier it was explained that Apple uses Xcode for their applications. So then if the project’s application was needed to be compatible with Apple devices, it would need to be rewritten in Xcode. Given how Android devices are much more common in today’s market, and this project is working with a rather small budget, there is not much justification for developing the app for anything other than Android. It may be a future consideration to make other versions for future expansions of this project, but for now the project will only be focusing on the Android market.

Many of the Android apps available today are made using the Java programming language. It is official supported for Android, and it is one of the top languages used for app development [4]. Java is taught at many universities for computer programming. In fact, most of the project team has some level of experience with Java thanks to some of the courses that have been taken at the university. This includes courses such as Object Oriented Programming, Computer Science II, and Processes for Object-Oriented Software. Additionally, the user interface of the application is created using XML. This language has some similarities with HTML, which most of the team are already decently familiar with.

If the project team had decided to create the app for other devices, such as Apple devices, it would require learning completely new programming languages in a small amount of time. In the case of Apple, this would mean trying to learn Xcode. While Apple does provide plenty of resources and tutorials for Xcode, it would still take a considerable amount of time to learn what is needed for the project. In contrast, the project team is already familiar with the programming languages used most commonly for Android app development. So sticking with Android for the creation of the app goes well with the team’s current skill sets.

Android also has a lot of features and components that are useful for parts of this project. As an example, it has the ability to add maps into an app for various functions [5]. This will be used to set up the map in the application, which will then help track the watch. The map is similar to Google Maps, so for a lot of users it should be familiar to work with. Having a map that users should already know how to use will help with the application’s usability. Android also offers support for Bluetooth [6] and other location strategies [7]. Though the project team ultimately chose not to use Bluetooth, it was still great to see that Android has compatibility with Bluetooth. Besides this, some of the other location strategies will be helpful in allowing the app to receive and properly display the watch’s location.

Another helpful component is the ability to let the app give notifications. These notifications are the messages that an app can display on the user’s phone, outside of the app itself [23]. This will be useful in letting the user know when new alerts have been sent from the Wander Watch hub to their app. Of course, Android also has the ability to allow apps to send and receive text messages. This will be part of the alert system for the application, to ensure that the app will be able to receive and display alerts from the hub. All in all, Android already has a lot of pre-built components and features that are needed for the project’s application, particularly for its major functions. By choosing Android, it will help ensure that the app’s development with be quick to complete.

In addition to Android and Java’s useful features, at least one of the team members is already experienced with developing Android applications used Android Studio. This is the official IDE for Android applications. In COP 4331 Processes for Object-Oriented Software, they were a part of a group that developed a small app for the class using Android Studio. If the project team were developing the application for other devices, then it would be necessary to learn how to use other IDEs for different kinds of apps. For example, Apple has an entire toolset and IDE exclusively for Xcode [24]. Much like the concern with spending too much time learning a new programming language, it was decided that trying to learn a new IDE for app development would be more trouble that it could be worth. Using an IDE that the project team already has familiarity with will help in developing the application in a timely manner.

The last major factor is that both Android and Java have a lot of tutorials available freely online. While many of the project team members have learned a lot about Android and Java programming through their classes, there are still plenty of topics that the team does not know a lot about. Fortunately, the tutorials for both Android and Java will help fill in the holes for this project’s success. These tutorials cover topics such as working with Android Studio, how to include components such as maps and alerts into an application, and how to build a basic app that can serve as the foundation for more advanced apps. Through their research for this semester, the Wander Watch team have already taken note of several tutorials and articles that will be helpful for this project.

For Android, Google has an extensive website dedicated to helping people with Android development. It even features a video course for getting started with building an app [8]. Meanwhile, the Java programming language has a large amount of documentation available on Oracle’s website [9], in addition to the various tutorials that are available elsewhere online. Since Android is the most widely used smartphone OS at the time of this writing, it makes sense that there would be plenty of resources like this for all the facets of Android app development. In comparison, other devices have their own sets of resources and tutorials as well. However, since they are not as widely used as Android, these resources are not as extensive or widespread for devices such as Apple’s iOS devices. Having a wide range of resources and tutorials will allow the team to learn more about Android and Java, and how to better apply many tools and features to improve the application for this project. This is why Android and Java are very easy to get started with for school projects such as this, and how Android and Java can allow people to become better app developers.

Ultimately, it was found that choosing to develop an Android app for the Wander Watch project is the best choice. Besides the cheaper cost, Android is the most popular smartphone OS in today’s market. This will ensure that the Wander Watch can be used by a large range of users. Focusing only on Android will help the team avoid spending unnecessary cost and time on developing different versions of the app for devices that do not have as large of a user base as Android. Not to mention, most of this project’s team are already familiar to Java programming and Android development. If there is something that is not known, the project team can easily refer to the many tutorials and resources available online for help. These reason are why Android was ultimately chosen as the development platform of the Wander Watch’s application.

5.2 Features

One of the main feature of the app is that users can log in with their own account. This account will be associated with their particular watch and hub. While setting up the account on the application, the user should have the phone close to the hub and watch that they want to use. This will help associate that particular hub and watch with the user’s account on their phone. After this, the user will be able to log into the account to check the status of the watch. They will be able to view the watch’s location, check any alerts they have received from the hub, and change settings associated with the application. The following pages will be discussing the appearance of the app and how it will work.

5.2.1 Menus

When the app is opened, for the first time, it will display a screen that will help the user through the process of setting up their account. The user will associate their account with a certain e-mail address, and create a password for their account. Then, the application will tell the user how to connect their particular hub to their app account. The hub will have to know the user’s account. Once it knows which account to connect to, it will set up its connection to the hub. It will sent a test alert to make sure that the user can receive alerts on their phone. When the user confirms that they can see the alert, the application will let them know that the setup process is complete, and will now display the main menu.

When the user opens the application after the setup process, it will display the login screen. Once the user logs in using their account email and password, they will see the main screen of the application. There are three main options on this screen: the Map, Alerts, and the Settings button. Clicking on each button will take the user to their respective menus. The following diagram shows how the user will be able to navigate between the application’s different menus.

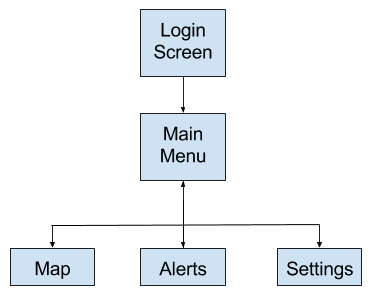


Figure ?.? - Navigation diagram of the application’s menus

These menus will be designed to be user-friendly. This includes making them easy to understand and being visually appealing. The user will also be able to change the appearance of the application in the Settings menu. The following diagram shows a mockup of what each menu will look like.

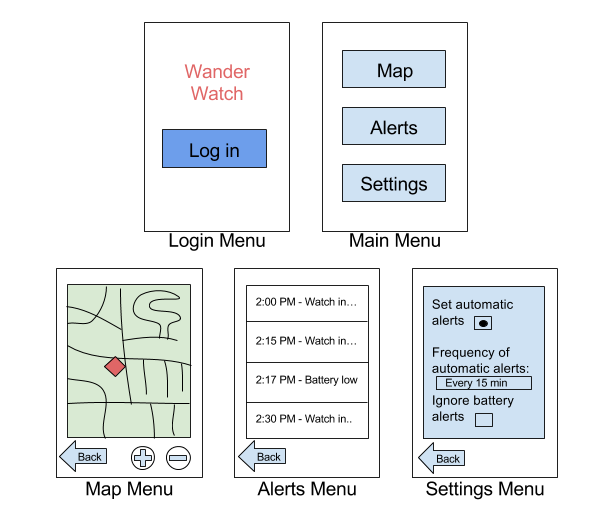


Figure ?.? - Mockups of the application menus

The final appearance of the menus may be different from what is shown here, but it will be ensured that the menus will stay simple and easy to use. Through the Settings menu, the user will be able to change the appearance of the application along with changing settings related to the alerts. This will be discussed in more detail in the next section.

The choice to have these three menus was made to make sure that the app’s functions could be organized into three easy-to-understand categories. It has been determined that these categories cover all of the application’s needed functions. At this point in time, the application does not need more menus or content than these three menus. However, it is possible that future expansions could add more functions that would require more menus to keep it properly organized. For the time being, these menus are sufficient for helping the user within the current scope of the project: keeping track of just one watch and one hub at a time.

5.2.2 Functions

Each menu is associated with a particular set of features and functions. They are named such that the user should be able to tell which menu will be related to a certain activity. So then, the Map menu allows the user to view the app’s GPS map, the Alerts menu displays alerts from the hub, and the Settings menu contains a list of settings that the user can change to meet their needs.

The Map menu shows the map which will track where the person wearing the watch is located. As mentioned earlier, this will be displayed as a standard Google Map implemented into the Wander Watch application. The user will also be able to zoom in and move the map around, much like the standalone Google Map app.

Like the standard Google Maps application, the map that will be shown in the Wander Watch application should be accurate within 10 to 30 feet. This will ensure that tracking down the person wearing the watch should be as effective and accurate as possible. At this point in time, the Wander Watch team does not plan to add anything extra to the Map menu. There many be more advanced or improved map features that can be added in future versions of the application.

The Alerts menu will show the user any alerts they have received from the Wander Watch Hub. As shown in the mock up, the alerts are shortened, as indicated by the ellipses that end each alert. When the user clicks on an alert, it will expand the alert so that the user can read the full message. These alerts can tell the user if the watch has left the hub radius, and at what time. This type of alert will inform the user to check the map for the person’s location, and to take steps to relocate the person. They will also be able to inform the user when the watch needs to be recharged. The Alerts menu will show up to the last 10 or 20 alerts.

The alerts themselves will also show a notification outside of the application itself. Many Android apps are able to display a small notification to inform the user that something has changed in the app. This happens even if the app is closed. These notifications will also be enabled for the Wander Watch app. This way, the user will know when they have gotten a new alert, and they do not have to keep checking the app themselves.

The Settings menu will allow the user to change certain aspects of the application. Here, the user can choose if they wish to receive automatic alerts. These alerts will be sent to the user at certain intervals. The user can also set when they want to receive the alerts. This could be every 15 minutes, every half hour, every hour, and so on. In addition, the user can chose to ignore alerts about the watch’s battery getting low.

Besides changing the settings related to the alerts, the Settings menu will also let the user change the color scheme of the application. This is a relatively simple function to program into an Android application, and will not take much time to implement. The options for the color scheme change will be such that the colors are not too harsh on the eyes. So this would mean no extremely bright colors, or color combinations that would cause the text to be hard to read.

The Settings menu will also give the user the option to change which hub and watch is associated with the account. For the scope of this project, each account will only be able to handle one hub and one watch. The user can also delete their account, which will remove all information related to their account, watch, and hub. They can create another account, but would have to redo the process of associate the hub and watch with their application account.

These functions cover everything the user would need to utilize the app and be able to keep track of their loved ones with the Wander Watch. Future expansions of this project could include more functions, such as the ability to keep track of more than one hub or watch. This will be further discussed in a later section.

6.3 Application Design

The following sections will be going into greater detail on the construction of the Android application itself. Here, the discussion will be focused on how it will be built, how it will be able to communicate with the hub, how the navigation of the application with work, and explain the parts that will be used to build up the application. No code will actually be shown here, but these sections will be going into details on the functions and implementations that will be utilized to give the application certain features.

6.3.1 Software Goals for the Application

The software goals for the application are meant to be used as a sort of guideline to making sure that the application is capable of certain functions. Besides its major functions, which were discussed in a previous section, this section will go over other goals that the Wander Watch application should be able to meet.

For one, the application should be able to be built, tested, and made fully functional within the time given for this Senior Design project. The Wander Watch team has done research into the feasibility of the application, and have determined that it is indeed doable for this project. In fact, it has been found that past groups in this course have build similar projects. Thus, if other people were able to complete projects similar to the Wander Watch, then the project team should be able to finish the application along with the rest of the project on time. A more in depth discussion of the project’s timelines and milestones will be discussed in the respective sections near the end of this report.

A second goal is that the application should have no problems communicating with the other hardware that will be used for the watch and hub. Through the research for this project, it has been determined that the Android application should be compatible with the watch and hub in terms of sending messages and alerts between each other. The hub is able to send alerts via SMS messages, and the application can be designed to receive these alerts as SMS messages as well. Using SMS messages on a cellular network instead of other kinds of communication has been determined to be the best option in order to keep the cost of this project as low as possible.

Another goal is that it should be able to be built using mainly Android Studio. Given that Android and the Java programming language has a lot of implementations and features that will be useful for this project, Android Studio already has practically everything that is needed to build the functions of the application. At this point in time, the project team’s research has not shown that something that does not already exist as an Android implementation or Android tutorial would be needed for this project. Something may happen down the line that would require use of other tools besides Android Studio. However, the team has not found anything that would require relying on another program, nor has the research up to this point suggest that the project will need something outside of the team’s chosen programs and parts. So for the time being, the Wander Watch application seems to be perfectly doable with what the project team has decided to use so far.

A goal that has been discussed previously is that the interface of the application should be user friendly. It should be easy to understand and use. For now, mock-ups of the screens of the application have been created, as shown in a previous section. The screens will be modified slightly once the application is actually built, but the project team will be making sure that the final application is still visually appealing and easy to understand. If it is found that some parts of the application are confusing to use, they should be changed accordingly. The project team will use certain guidelines and benchmarks while testing the application to make sure that it is indeed user-friendly for most users. This will be discussed further in the Application Testing section.

One final goal is that the application should not take up too much battery power on an Android phone. According to the team’s research, it is believed that battery power will not be too much of an issue for this project. The Wander Watch application is relatively simple in design and function. Thus, it should not take up too much of the phone’s power. Regardless, the project team will still be aware of possible battery problems. The application will be tested for any such problems, and part of the testing phase will be used find ways to optimize the application so that it doesn’t use a lot of power. Taking time to further optimize the application’s power consumption will only be done if it is found to be necessary.

6.3.2 Communication with the Hub

The hub and application communicate with each other to share information on the watch’s location, and on the user’s settings for which alerts can be sent. Based on the user’s settings, the hub will send automatic alerts for letting the user know whether or not the watch is still within range, or to let them know when the watch is low on battery power and needs to be recharged.

These alerts will be sent to the application via SMS messages. Android applications are capable of receiving text messages via SMS communications, and there are plenty of texting apps that work by sending messages this way. Thus, it has already been proven that an application can send and receive these SMS messages. So as long as the hub is connected to the right cellular network, and it is associated with the right account, the user should be able to receive the alerts from the hub without any major problems. The following diagram shows these communications for the project.

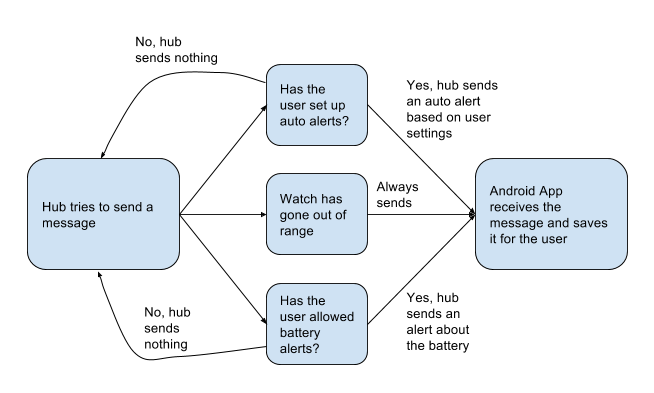


Figure ?.? - Communication flow diagram

As discussed in the Functions section, the application will only receive messages based on how the user has changed their settings. If the user has not set up automatic alerts, the hub will be notified that it should not send these kinds of alerts. Likewise, the hub will not send alerts about low battery power if the user does not allow it in the settings. The only kind of alert that will always be sent are those that indicate when the watch has gone out of range.

The need for a cellular network connection does present a problem, though. The hub needs to always have a strong connection to a cellular network, and the phone that has the application also needs to be able to access such a network, or otherwise be able to receive SMS messages via a different type of connection. However, there is a large availability of cellular networks today, and these networks are continuing to grow in availability. Any phone should be able to connect to the cellular network needed for the application, and receive messages via this connection without needing additional hardware or software. Indeed, more and more users are relying on cellular networks as they continue to grow in availability in many locations. Even though SMS messages have a cost attached to them, most of them offer a fairly minimal cost for most users. In fact, for this project it has been estimated that each alert should only cost about ten cents. Thus, the project team feels that using these types of networks for communication will not be that much of an issue for the average user.

The Wander Watch team had originally considered simply making it so that the hub sends a text message to the user’s phone, so that the user does not have the need to use a separate app. However, these text messages are sent through the phone’s carrier network. This could involve needing permission from various carriers in order to authorize the message. This in turn could have limited how many people would be able to use the Wander Watch, depending on what carrier would let the project team use their networks for the alerts. This is a similar case to one of the previous products found during the team’s research, the Comfort Zone Check-In. That product could track a person via their cellphone, but only if it was already on the Sprint network. Besides needing permission from various cellular networks, this could also incur a cost for being able to send the alerts through these networks. This could very well put a large cost on this project’s users. The team wants the Wander Watch application to be able to work without being limited to certain networks like this. The particular cellular network that was chosen for the project was determined to be the cheapest per message sent, while also been one of the largest networks available. It was decided that this was the best option to make sure that this project will work in a wide range of locations, while still making it affordable to build and use.

6.3.3 Code Flow

Code flow refers to how a user moves through the program while said program is running. In this case, the program is the Wander Watch application, and the code flow describes the possible ways the user can navigate through the app. In the Menus section, it was discussed how the user can move from one main menu to another. This section will go into detail on the navigation within each menu.

This is essentially a more in depth version of the menu navigation diagram shown in the Menus section of this report. Besides the overall navigation between each major menu, code flow analyzes how different decisions can be made based on the information the user have given to the application. These decisions can determine what kind of information the user will be able to see, and where they can go within a menu.

The following flow control diagram shows the possible ways that the user can move from one part of the application to another, including within a menu.

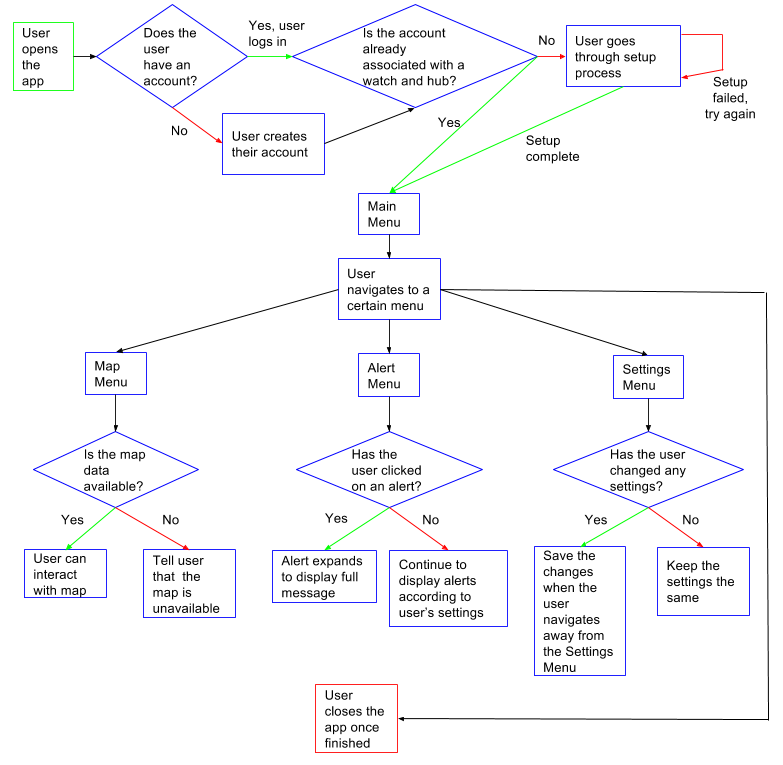


Figure ?.? - Control flow diagram for the Wander Watch application

The control flow diagram shows other processes that the application will need to handle for navigation. Overall, the application will be able to check five major interactions. First, if the user has an account. Second, if the user has gone through the setup process to associate their account with a specific hub and watch. Third, if the map data is available to be displayed within the Map Menu. Fourth, if the user has clicked on an alert in the Alert Menu. Fifth, if the user has changed any settings in the Settings Menu.

This diagram also includes situations where the application may fail a certain process, or simple not complete a process due to the user changing the settings for their account. For example, if the set-up process fails for whatever reason, the application will prompt the user to try the process again. Also, the application may be unable to retrieve the map data for the Map Menu. This could be due to a bad connection, either from the phone or the watch for whatever reason. If this happens, the application will tell the user that the data is currently unavailable, and will not display a full map.

The application will also check for changes in the Settings Menu. Instead of having a dedicated button for saving, the application will check for changes when the user navigates away from the menu. If the user has made changes, the application will automatically save the changes before navigating back to the main menu. The user will be able to tell that the changes have been saved by seeing a small “Saving” prompt before the navigation is complete. If the user has not made any changes, the application will continue to the main menu without trying to make any changes to the user’s settings.

The user should be able to exit the application from any of the menus, even from the login screen. This control flow diagram shows the navigation and if-else decisions that will be taken into consideration for the application’s actual code. This also helps the team figure out the kinds of Java functions and code that will be needed to ensure that the application can navigate properly between the menus, and complete certain actions based on what the user does to their account.

6.3.4 Diagrams and Structures

The following two sections will discuss some of the inner workings of the application. The first section will discuss the application’s class diagrams. This shows some of the actual Java functions and code that will be expected to be used in building the application’s functionality. The second section will cover the data structures that will be used for this project, and their relation to the application. This means relating the different pieces of data that the applications will utilize with the parts of the application itself.

Both of these diagrams may be modified as work continues towards the final product of this project. Still, these diagrams can represent, based on the research for this project and progress on the project so far, an estimate of what can be expected to find within the final application for the Wander Watch project.

6.3.4.1 Class Diagrams

There are many different functions that will be programmed in Java to ensure that the Wander Watch application will work correctly. Many of these range from simple to complex, depending on what the application needs to do. The following class diagram shows the classes and Java functions that are expected to be utilized in the application. This may change in the future, either by adding new classes and functions, removing those that are found to be ultimately not needed, or replacing an old class with an entirely new class. Thus, this diagram is not final, but based on what is needed and what the application needs to do, the project teams has estimated that it is a decent representation of the classes and functions that will be found in the final application.

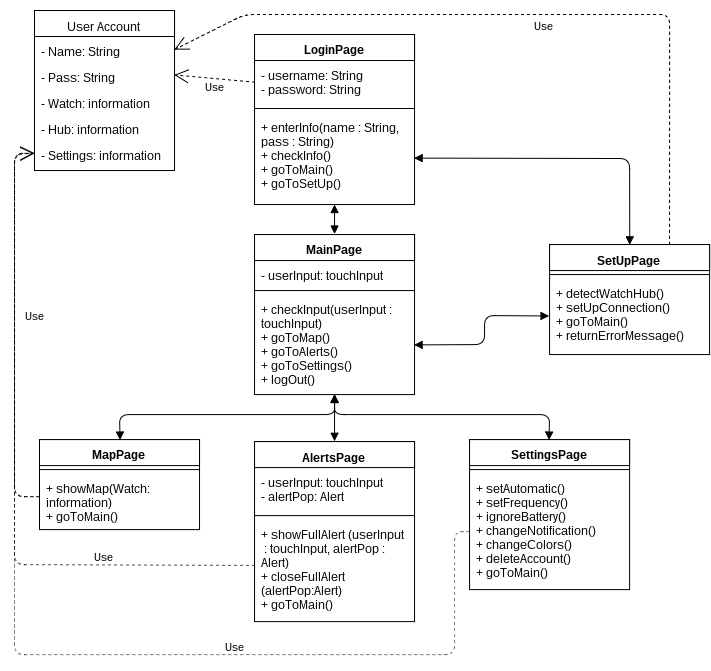


Figure ?.? - UML class diagram for the Wander Watch application

The user’s account acts as an overall container of information. It has info on the user’s username, password, the watch associated with their account, the hub associated with their account, and the settings for their account. The pages for login, setup, map, alerts, and settings all use information based on the data saved in the user’s account. Starting at the login screen, the user inputs their username and password. This is processed by the enterInfo function, which in turn will let this information be checked by the checkInfo function. The application checks the inputs using the checkInfo function, and navigates to the main screen if the information matches with what is in the user’s account.

If the user does not have an account already, they are redirected to the setup page, to set up their account and the connection with their watch and hub. On that page, the detectWatchHub and setUpConnection functions are used to find the user’s watch and hub and establish a connection between them and the application. If an error occurs, the application uses returnErrorMessage to let the user know that something has gone wrong, and will prompt them to try again. Otherwise, the user will be sent to the main page via the goToMain function once the setup is completed.

The main page contains functions to help the user navigate to other pages. It collects information on where the user has clicked, hence the userInput variable. From there, the user can navigate between the map, alerts, and settings screens. This is based on which button they have clicked on. Each navigation function corresponds to their respective page. For example, goToMap will go to the map page. The user can also log out of their account from the main page, since the logOut function is contained on the main page.

On the map page, it uses the function showMap to get information about the watch’s location, and display an accurate map of where the watch is. The watch information is stored in the user’s account information. So this function will get the necessary location data from the user’s account. On the alerts page, it collects information about the user’s alert from their account and displays these alerts. The userInput function is again used to display the right data based on where the user clicks on. For this screen, the user taps on a particular alert to display its full information, or to close an already opened alert. It has the showFullAlert function to allow the user to show a full alert, and the closeFullAlert function to close it. The alertPop variable holds the full message of the chosen alert. Thus, alertPop will be used to display the full message when the user chooses to open a full alert.

On the settings page, there are functions to change a particular settings based on how the user interacts with its corresponding input. The setAutomatic function will set automatic functions for the user. The setFrequency function is used to set up the frequency of which the automatic functions will be sent. So if the setAutomatic function is changed so that the user will not receive automatic alerts, then the setFrequency function will do nothing. The changeNotification function is related to the notifications that will appear outside of the app. The user can choose to allow or prevent these application notifications. The changeColors function will change the color scheme of the application itself, and is purely a cosmetic change. The deleteAccount function will of course delete the user’s account information from the application. If the user choses to delete their account, then all of their information will be wiped, and they will be sent back to the log in page to make a new account.

All three of these pages have the goToMain function to allow the user to navigate back to the main screen. Since only the main page has the logOut function, the user can only log out from there. However, they will be able to close the application on any screen, just like any other Android application. There may be more functions or classes added to the application, but for the time being this class diagram covers most of the functionality that is needed for this project.

The functions themselves will be built in Android Studio using mostly Java. The code used within the functions has not been specified, since at this point it is not known what code will be needed for the functions to properly work. However, giving what is known to be necessary for the application’s success, it should be simple to understand what kind of code, loops, and other functions may be needed to ensure proper functionality for the application.

6.3.4.2 Data Structures

There is a lot of data that the Wander Watch application will work with in order to function properly. This not only includes the GPS location of the actual watch, but also information regarding the user’s account. The following data structure diagram shows a visual representation of the kinds of the data that the application will have to deal with. It also shows what parts of the application will have to process which type of data.

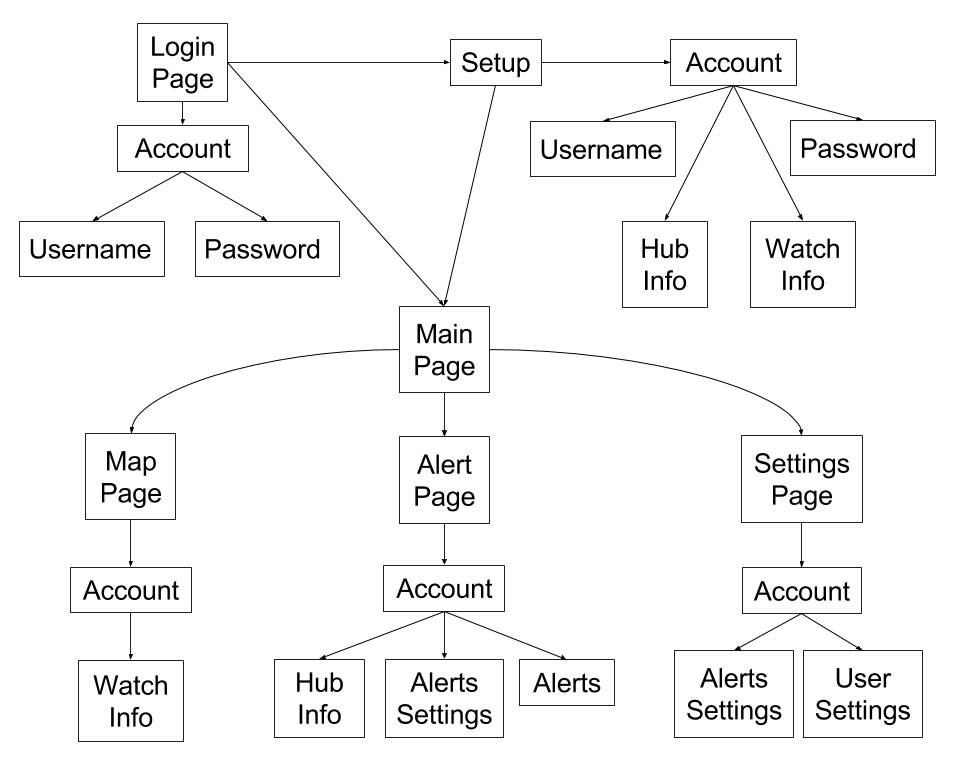


Figure ?.? - Data structure diagram for the Wander Watch application

As stated in the Class Diagram section, all the information that the application needs is associated with the user’s account. So in order to get the right information, each part of the application needs to access the user’s account and make sure to retrieve the right information. Note that the alerts settings and user settings are not altered during the setup process. These settings will be set to their default values on a new account. The user can go to the settings menu to change these settings to their own preferences. Similarly, the user’s account will not have any alerts associated with it when it is first created. Once the user starts receiving alerts from the hub, these messages will be stored alongside their account information.

The login page checks and verifies the user’s username and password. The setup page will populate the information for the user’s watch and hub during the setup process. From there, the other pages use parts of the account to perform their functions. For example, the map uses the watch information from the account to show the user an accurate map.

It is a good idea to keep all the information associate with a single account like this, because it helps to keep this information organized and easy to find. The information is stored on the user’s phone, rather than stored on a separate server or hard drive. This way, the user can change settings without needing a connection to a server. This also allows the application to store information about the watch and hub in an easy to access place. This in turn will allow it to get the right info it needs to display the right map, and the correct alerts from the hub.

8.5 Application Testing

Since there will be testing phases for the watch and hub, there will also be one phase exclusively for the Wander Watch application. The application testing will be focused on ensuring that the application works as intended, and that it is easy for most users to understand and use.

The first phase of testing will be centered around the application’s features and functions. The major goal is to make sure that the application can track the watch via GPS, and that the application can also receive and display alerts from the hub. Since these are the two major features of the application, it is vital that these work before any superficial or cosmetic features that the project team want to add in afterwards.

First, the tracking feature will be tested. Once the map extension has been added to the map, it will be ensured that it actually displays a map, and that the map itself displays accurate location information for the phone itself. If the map is not displaying properly, then the project team should be able to look into it and figure out how to fix it. From there, the team will work on implementing the tracking function so that the map will display the location of the watch rather than the phone that has the application. The map extension that is available from Google will usually display the phone’s own location by default. The project team will need to ensure that the application can receive the information it needs about the watch’s location from the watch itself. So the team will be testing not only the application’s ability to receive the location data, but also the watch’s ability to send the data to the application in the first place. It is expected that this will provide some trouble in getting it right, but the project team has made sure that there will be help available for the project. This help will be expected to come from professors here at the university who are experts in device to application communication.

Second, the project team will be testing the application such that it can properly receive and display alerts from the hub. Before the team tests it with actual hub alerts, the application will be tested with regular text messages. The team’s research has found tutorials about making text messaging Android applications that are able to send and receive messages from one another. This tutorial will be the basis for something similar to test the application’s ability to receive messages. At the same time, the application’s ability to display notifications will also be tested. Notification are alerts or updates that are displayed on the phone, but outside of the application. This is so that the user can know that something has happened without needing to constantly have the application open. Once it has been determined that the application can both receive messages and display notifications, the project team will start testing it with actual messages from the hub. Since Android applications can already be built to receive text messages as SMS messages, the project team will be handling the hub messages in a similar manner. The hub will be tested to send the alerts as SMS messages, and the applications will be tested to receive them. As with the GPS testing, the project team has already kept a few professors in mind to ask for help if problems with the hub alerts occur during testing.

Overall, it is expected that this part of the testing will take at least 2 weeks to complete. This testing will be performed while testing the watch and hub as well. It has been determined that this would not only save time, but also help the project team find problems or bugs faster. In addition, since the watch, hub, and application depend on each other for information, it is important that all of three of them are tested together to ensure smooth operation and to fix any problems as soon as possible.

The next part of the application testing will be focused on making the application user friendly. For this part, the project team has decided to have some outside testers try out the Wander Watch application. The team will be observing how the testers get used to the application, and taking note of any problems or confusion they may experience. This also includes keeping track of any bugs or crashing that may occur while the application is in use. If any of these occur, the Wander Watch team will take steps to look at the code and fix these bugs as soon as possible.

Though user friendliness tends to be an objective part of any application, the project team wants to make sure that the application is easy to use and understand for most users. The team will measure the user friendliness of the project’s application by asking the testers how they feel about the application. This will mostly be concerned with whether or not the testers had any trouble logging into and navigating the application, if they had any problems receive alerts, if they had any trouble viewing and zooming into and out of the map, and if they found the settings to be usable.

The settings themselves will also be tested for any bugs or problems. The team needs to make sure that turning battery alerts on and off still works, and that automatic alerts can be properly set up and received. The project team will also test if changing the colors of the application works, and if the available color combinations are not eye straining for most users. However, this particular feature is one that is low on the project’s priority list, seeing as it is superficial and does not significantly impact any of the application’s more vital functions.

Once all of this has been tested and ensured that it works as intended, the project team would like to possibly add in and test new features for the application, if there is enough time to do so. Such features include the ability to connect the application to more than one watch and receive alerts from more than one hub. If the team works to implement these features, then said features will have to be tested to receive alerts and location information from all of the new watches and hubs. For location data, the application would need to have the ability to switch between different maps, with one map being for one particular watch. For the alerts, the application would need to have a good organizational scheme for the alerts. On the alert screen, the application could have it so that the user can see the alerts for each hub or watch, and slide left or right to view alerts from a different hub or watch. In turn, new features and functionality like these will have to be tested extensively for bugs. This is especially important because, since the application will be receive even more information and data, it could be difficult to manage and properly display everything.

However, this may be beyond the time and scope the Wander Watch team has for this Senior Design project. As a result, any further features such as these may not be worked on during the time for this course. It is still good to think about future expansions in case the team gets the opportunity to further improve the project. Even so, the team plans to focus on testing the GPS map, receiving alerts, and usability of the application. Anything else would be great to add on in addition to the major features, but the project team would not consider them necessary for the completion of the project’s current scope.

9. Administrative Content

The following sections will go into detail on non-technical parts of the project, such as how the project will handle its budget and the time spent on the project. Specifically, the first section will go over the budget and finance of this project, while the second will go into detail about the project’s expected milestones and timelines. The first section is meant to show the research made on the amount of money needed to buy the right parts for this project. It will also go over other finances that need to be taken into consideration for the project. The second section is to show how far the project team has planned ahead, and how the team has organized the time that is expected for building the watch, hub, and application. It will show how the team plans to complete the Wander Watch project in a timely manner, and devote enough time to finish each major section of it.

9.1 Budget and Finance

Having a good budget is an important aspect of any good project. It is necessary to keep track of how much money is being spent on what parts or services. This section will explain how much money the project team expects to spend on parts and other items needed for the project. It will also help show how they plan to make it so that their project is easy to build and produce.

The following table has been made based on the project team’s research into how much each component of the Wander Watch should cost. Each part shown here has been determined to be necessary for the project to function as planned. The application itself will not cost anything on its own. Most of the cost lies in the physical parts that make up the watch and hub.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **List of Final Parts** | | | | | |
| **Part Type** | **Name** | **Part No.** | **Manufacturer** | **Quantity** | **Unit Price** |
|  |  |  |  |  |  |

Table ?.? - Project budget

It should also be noted that the messages will be sent over SMS, so this will incur a cost depending on the network. However, as stated in previous sections, this will only be a very minimal cost, up to roughly ten cents per message. This is based on the $3 a month fee for thirty messages, which the team found was the best price that could be found for SMS messages. Overall, the project team expects the Wander Watch project to cost [TOTAL HERE]. This total includes both the parts and the monthly fee for messages for one month.

The Wander Watch team has been unable to secure outside funding for our project. Originally, Texas Instruments was considered as a major sponsor, but the team could not show how the project could be accepted for TI’s funding program. The team has been unable to obtain funding from other groups well. Mainly this was due to how the purpose of the Wander Watch relates mostly to taking care of dementia patients and the elderly. This is a useful application, but it does not fit the interests of most Senior Design sponsors. Many of the sponsors the Wander Watch team has spoken to were mostly interested in projects that could be used for various military applications. Others were looking for projects that could be utilized for expanding their own existing products and projects. Some groups had already given their sponsorship to other Senior Design teams by the time the Wander Watch team tried to make contact with them. As a result of this lack of sponsorship, this project will be funded with the team member’s own set of funds.

Since a major goal of the project is to make the Wander Watch affordable to produce and purchase, the team will be using this opportunity to show that this project can be doable on a low budget. Other projects may take hundreds of dollars more than what is planned for the Wander Watch project. Some projects may even cost thousands of dollars. The team has been focused on keep the cost of this project down as much as possible. This is why the final list of parts has parts that are low in cost but will still satisfactorily meet the requirements for the Wander Watch.

9.2 Milestones and Timeline

This section will discuss how the project team will plan out the timeline for building and testing the watch, hub, and application for this project. It is important to budget both money and time wisely, so that the project can be completed on time. If everything is done early enough, the project team may have extra time for additional testing and bug fixing. This extra time could also possibly be enough to add in extra features. However, even if the team does not have enough time to add more features, it is expected that the team will have enough time to at least finish what the project needs to function correctly.

9.2.1 Major Milestones

For this project, a milestone is considered to be a point at which the project team completes a significant step in developing and building this Senior Design project. The first major milestone will be done by the end of the fall semester. This will be the conclusion of the team’s research, which will show that the project team is now prepared to move on to building the watch, hub, and application for the Wander Watch project.

The project team has also begun collecting the physical parts for build the watch and hub. All of the parts that are needed will be collected before the beginning of the spring semester of Senior Design. This way, the team can have plenty of time to begin putting together and testing these parts. This will also give the project team enough time to change parts or obtain more parts in case something goes wrong during testing or building the watch and hub.

Concurrently with the collection of the physical components, the project team will also make sure that the basic skeleton of the application is developed before the beginning of the spring semester. This skeleton will have the basic features of the apps. These include: a working login screen, users have the ability to create an account and log in with it, and having the main screens, which are Main, Map, Alerts, and Settings. These screens will be fairly basic. For example, the Alerts screen will not have any actual alerts, it will probably have some test messages in place of actual alerts. Likewise, the map will not display the location of the watch, since the watch will not be built and tested at that point. Instead, the map will just display the current location of the user’s phone. In essence, this will be similar to the expected final application, simply without the correct map, ability to receive alerts, and ability to change settings related to the alerts and watch.

So by the beginning of the spring semester, the Wander Watch team will have three major milestones completed. Once the spring semester starts, the team plans to build and test the three aspects of this project as soon as possible. This will be done so that this project can be finished as soon as possible, while also giving the team plenty of time to test everything and fix any problems. The building of the watch, hub, and application are each their own milestone. The testing of each will also likewise be separate milestones. Ideally, the building and testing of each of the three aspects will happen simultaneously. Since each aspect relies on some features of the others, having them worked on at the same time will help with testing communication between the three. This not only saves time for building and testing, but will also help the team find any bugs or problems as soon as possible.

The final planned milestone will be the presentation and demonstration of this project at the end of the spring semester. This will show that the project team has completed a successful project for Senior Design. It will be ensured that the building and testing of this project will be completed well before the demonstration. If the team spend too much time testing, the project could end up being rushed, thus finishing the Wander Watch project with a lackluster project.

In order, the project team has 10 major milestones for this Senior Design project. These are finishing research by the end of the fall semester, collecting all the necessary physical parts, building the basic skeleton of the application, building the watch, building the hub, adding the required features for the application, testing the watch, testing the hub, testing the application, and giving the final presentation and demonstration of the project.

Preferably, the Wander Watch team will be able to reach these milestones on time. To aid the team in keeping track of what they should work on, an estimated timeline of the project has been created. This timeline is shown in the next section. It will help the team stay organized and focused on what is needed to complete each major step of the overall project. In addition, it will help the team keep track of how much time is spent on certain steps, and whether or not the project team can afford spending more time on later steps, or have to speed up in order to finish everything by the end of the spring semester.

9.2.1 Planned Timeline

The following timeline is only a rough estimate of how the Wander Watch team expects to spend time for the project. Milestones that are closer to the current date are more exact, but those further into the spring semester may be subject to change. This will depend on how much time is spent on reaching a particular milestone, or if certain problems and delays occur along the way. The later dates, such as those for building and testing each aspect of the project, are the latest date that the project team expects to finish with a particular step. In essence, each milestone should be completed before the latest date listed here in this timeline. The final date for the presentation will change based on the actual date that will be set for the spring semester. The data listed on the timeline is based on the last date of the official final examination week for the University of Central Florida for this semester.

Overall, the plan is to finish all the necessary building, implementation, and testing during the spring semester. The best case scenario is that the Wander Watch team will able to complete these steps well before the final presentation. Besides allowing for more time for testing or additional features, this extra time should also allow the team to properly prepare the final presentation and demonstration. It is important to do so as soon as possible since the presentation will not only show how much the team has learned over the course of this project, but also whether or not the project was successfully completed. The project timeline is shown below.

|  |  |  |
| --- | --- | --- |
| **Project Timeline** | | |
| **Date** | **Title** | **Description** |
| 11/12/15 | Research Completion | Finishing this essay, which means completing all necessary research and planning for the project. |
| 1/11/16 | Complete Parts Collection | Finishing collecting and buying all the necessary physical parts for the watch and hub. |
| 1/11/16 | Complete Skeleton Application | Finishing the basic skeleton of the Wander Watch application. |
| 2/29/16 | Building the Watch | Implementing the physical parts of the watch. This will include the casing and strap for the watch. |
| 2/29/16 | Building the Hub | Implementing the physical parts for the hub. This will include the casing for the hub. |
| 2/29/16 | Adding Major Features to Application | Implementing the major features of the application, to allow for communication with the watch and hub. |
| 4/30/16 | Testing the Watch | Testing the watch so that it fulfills its intended functions, such as sending location information to the application. |
| 4/30/16 | Testing the Hub | Testing the hub so that it fulfills its intended functions, such as keeping track of the watch. |
| 4/30/16 | Testing the Application | Testing the application so that it fulfills its intended functions. |
| 5/4/16 | Final Project Presentation and Demonstration | Presenting and demonstrating the final complete project, to show that all the aspects works as intended. |

Table ?.? - Project Timeline

It is hoped that this timeline will help serve as a reminder about what the project team should be focusing on. It should also aid them in keeping track of what needs to be done, and how much time can be afforded to the completion of a certain milestone.

(check the sources of sources (i.e. don’t use Wikipedia, but use its sources!))

Sources

[1] <https://developer.apple.com/support/compare-memberships/>

[2] <https://support.google.com/googleplay/android-developer/answer/6112435?hl=en&rd=1>

[3] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

[4] <http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2015#index/2015/0/1/0/0/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/>

[5] <https://developer.android.com/training/maps/index.html#start>

[6] <http://developer.android.com/guide/topics/connectivity/bluetooth.html>

[7] <http://developer.android.com/guide/topics/location/strategies.html>

[8] <https://developer.android.com/training/index.html>

[9] <https://docs.oracle.com/javase/tutorial/>

sources related to similar products

[10] <http://www.alzheimers.net/8-8-14-location-devices-dementia>

[11] <http://www.projectlifesaver.org/enroll-a-loved-one/>

[12] <http://www.mindme.care/>

[13] <http://www.gpssmartsole.com/>

[14] <http://www.gpssmartsole.com/ble-smart-sole.php>

[15] <http://www.safelinkgps.com/index.php/get-started/how-it-works>

[16] <http://pocketfinder.com/gpsseniortracker/>

[17] <http://www.revtrackbuy.com/>

[18] <http://www.comfortzonecheckin.com/default.aspx>

[19] <http://www.bluewatersecurityprofessionals.com/elderlytracking.htm>

[20] <http://www.alzheimer.ca/en/Living-with-dementia/Day-to-day-living/Safety/Safely-Home>

[21] <http://www.keruve.com/best-personal-locator/>

[22] <http://www.ohwr.org/projects/f-watch/wiki>

[23] <http://developer.android.com/guide/topics/ui/notifiers/notifications.html>

[24] <https://developer.apple.com/xcode/download/>

[25] <http://www.androidauthority.com/qa/question/sending-receiving-sms-messages-via-wifi/>