

Context-Aware Clothing Recommendation

Project Plan by:

William Parr

Tyler Witte

Ethan Wiczorek

Nickolaus Eaton

Christian Ehlen

Nicus Hicks

1	Introductory Material	4
1.1	Acknowledgement	4
1.2	Problem Statement	4
1.3	Operating Environment (one paragraph +)	4
1.4	Intended Users and Intended Uses	4
1.5	Assumptions and Limitations	5
1.6	Expected End Product and Other Deliverables	5
2	Proposed Approach and Statement of Work	6
2.1	Objective of the Task	6
2.2	Functional Requirements	6
2.3	Constraints Considerations	6
2.4	Previous Work And Literature	7
2.5	Proposed Design	7
2.6	Technology Considerations	8
2.7	Task Approach	8
2.8	Possible Risks And Risk Management	9
2.9	Project Proposed Milestones and Evaluation Criteria	10
2.10	Project Tracking Procedures	11
2.11	Expected Results and Validation	12
2.12	Test Plan	12
3	Project Timeline, Estimated Resources, and Challenges	14
3.1	Project Timeline (Will)	14
3.2	Feasibility Assessment	15
3.3	Personnel Effort Requirements	16
3.4	Other Resource Requirements	17
3.5	Financial Requirements	17
4	Closure Materials	18
4.1	Conclusion	18
4.2	References	19
4.3	Appendices	20

List of Figures

Figure 1: Task Approach Breakdown

Figure 2: Gantt Chart

List of Tables

Table 1: Breakdown of three main phases, including basic timeline, for the semester.

Table 2: Time requirements in detail for project deliverables, broken down into development time and testing/documentation time.

1 Introductory Material

1.1 Acknowledgement

Professor Goce Trajcevski will be assisting in the technical advice and equipment retrieval for this project.

1.2 Problem Statement

Sometimes it is difficult to predict what the weather will be later in the day, and someone could properly prepared for an earlier part of the day with the clothes they have on currently, but unseen changes in the weather will bring problems for this person. There is also the case of someone travelling for an extended period of time. Not being native to the area, a traveler will not have the same idea of what to wear at a location as the people who live there, so this project attempts to create a solution to these problems presented.

This project aims at providing a comprehensive solution for a clothes recommendation system. The main idea is to couple data from multiple sources and integrate it in a manner that will enable the users to select outfit and/or plan which clothes to pack for an expected trip. As a motivational example-scenario: upon detecting that the user has woken up (e.g., Fitbit), a trigger is fire that connects to a Weather Channel API. Given the weather prediction and the tasks and places that the user has entered in the calendar for that particular day, an app will provide a recommendation for selecting the items from the closet to be worn/carried for the day. In addition, in case of a multiple options, the hangers in the closet will provide another source for selection.

1.3 Operating Environment

This product will mostly be used on smartphones with some users potentially using a web version of the application. Operating environments will not really affect our application any more than the normal operating environments of the user's smartphone.

If we do end up moving to include a hardware aspect of the project, our hardware aspect will consist of a single screen hooked up to a computing device (a screen with a raspberry pi, or a cheap android tablet, etc.) that will live in a singular location in the user's closet. The operating conditions of the closet will be easily manageable for any computing device we decide upon as the device will be stationary and in a dark, dry location.

1.4 Intended Users and Intended Uses

Our first primary intended user is those who travel frequently for business. These individuals may be traveling to different countries in the duration of just one or two days. Our application is intended to allow these users not to worry about checking the weather, but rather just have it done for them. Our application will also take their schedule into account as well, so that it will give smart predictions based on the weather at that time of day and also where they will be.

Another primary user are those who live in an area where there is easy access to public transportation. These individuals may have a longer commute and may not be able to go home during the day. Therefore, our application needs to be able to give recommendations for an entire day based on the weather.

Our secondary user is those who would like to make their day more efficient. These individuals would like to have an outfit prediction based on the weather, and may not have a busy schedule. Picking out an outfit takes time, and our application will make their day more efficient.

1.5 Assumptions and Limitations

Assumptions:

- System will run on a single-user basis, so should be able to run on many concurrent systems.
- That the user will be able to have internet connection a majority of their time, and always when they are getting ready for the day.

Limitations:

- The system will need to be connected to the internet in order function.
- Users will need a smartphone in order to run the end product.
- The integrated development environment won't be available to us on school computers so work will take place off campus

1.6 Expected End Product and Other Deliverables

The only currently planned deliverable is the android, iOS, and web application. The application will have the ability to do all available functions across all 3 versions of the application. The client will be able to first input all of their owned pieces of clothing and catalog them in their digital "closet." The user will then set a daily time they would like to be initially told an outfit for the day (this will normally be the same time as their alarm). The user can also request an outfit update at any point of the day inside the application. The application will provide an outfit based on the current and possible weather for the day at the specified time every day. The application will suggest clothing that the user already owns and has catalogued

in the application. The user will be able to see alternative options for the clothing items suggested in case the top suggestion is dirty or not currently usable. The application will also periodically check in throughout the day to see if the weather predictions have changed and suggest items like snow boots, raincoats, and umbrellas. The application will notify the user if any updates that require a sudden change in attire do occur.

The application will reach the alpha stage by december 2018. The alpha stage will consist of an initial product we will use to retrieve feedback from testers and potential users. The beta stage will be when we have mostly finalized the product and are looking for improvements will be reached by May 2019, and the finalized application will be released three weeks before the end of the Spring 2019 semester.

2 Proposed Approach and Statement of Work

2.1 Objective of the Task

Describe the goal of the task. Depending on the type of project, the resultant end product can vary significantly:

- An actual hardware/software product – The design of a product
- A process to accomplish something
- A service to be performed
- A simulation or a set of calculations – Some combination of the above

The goal of this task is to produce a context aware application that will connect to a weather API and the user's personal calendar to produce a clothing recommendation from the user's wardrobe based on the data retrieved from the two sources.

2.2 Functional Requirements

The functional requirements of this project include the ability for the application written in ReactJS Native to communicate with an external weather API and the user's calendar. Once the data is retrieved from these sources, the application will contact the database that contains elements representing the user's wardrobe, and based on what clothing is available to wear at the time, the weather, and the events of the user's day, the application will display a list of all the different pieces of clothing that would be appropriate to wear for that day.

- Log in and out
- View available clothing
- Select clothing to be worn that day
- Add/Remove clothing from wardrobe
- Refresh wardrobe when wardrobe is low or out

2.3 Constraints Considerations

Non-Functional Requirements

- **Scalability** - The database of the application must be scalable to ensure many users will be able to access the application and their wardrobes.
- **Availability** - The application needs to be available 24/7 for when the users require context updates for their wardrobe.
- **Reliability** - Application must be able to recover loss of database and preserve user accounts and wardrobes.
- **Maintainability**- Database must be maintained to ensure proper updates via the weather and calendar.
- **Security** - Database must be secure to protect information about the users and their clothing.
- **Data Integrity** - The clothing items stored in the database should only be able to be modified by the users or an administrator.
- **Usability** - Will be accessible by all users and administrators, so all users can receive updates for clothing recommendations. Users will connect to the application via a mobile device and recommended clothing will be displayed as the weather and calendar are checked.
- **Performance** - The application should generate a recommendation based off the weather and calendar events in less than 3 seconds.

Standards

All development will take place remotely and outside of the lab environments, and we do not think there will be any practices within the development of this project that would be deemed unethical, but we will still adhere to the standards and protocols brought forth by IEEE/ABET. If we have any concerns about this, we will be sure to consult or client and advisor about possible breaches in the standards and protocols

2.4 Previous Work And Literature

There is a web application similar to this project called “Daily Dress Me”, and this application looks at the weekly forecast and makes clothing recommendations on the temperature and overall weather. The difference between “Daily Dress Me” and our application is that our project is going to be a mobile application rather than a web application. Another addition to our project that will separate us from “Daily Dress Me” is our project will also be making recommendations based on the user’s calendar, and the clothing recommended will be pieces contained in the user’s personal wardrobe rather than just a general outfit like provided on “Daily Dress Me”.

Another project that has a similar description to ours is a previous is an application designed by our client. This is an application that creates recommendations for public transportation for users based on the weather. This will be similar to ours in the sense that we will both be generating the best solutions for the user based off the weather situations, but they differ in the aspect that this application deals with the most optimal routes to take with public transportation based on the weather, while our project deals with preparing the user for the weather at the beginning of the day.

Source: <http://dailydressme.com>

2.5 Proposed Design

This project aims at providing a comprehensive solution for a clothes recommendation system. The main idea is to couple data from multiple sources and integrate it in a manner that will enable the users to select outfit and/or plan which clothes to pack for an expected trip. As a motivational example-scenario: upon detecting that the user has woken up (e.g., Fitbit), a trigger is fired that connects to a Weather Channel API. Given the weather prediction and the tasks and places that the user has entered in the calendar for that particular day, an app will provide a recommendation for selecting the items from the closet to be worn/carried for the day. If the proposed outfit is in the wrong temperature range for the user, the user can look at a suggestion in a higher or lower temperature. They can permanently set this as an option on their account if they are a person who tends to be cold or warm more often than the average we find.

2.6 Technology Considerations

The strengths of the technology we will be using is mostly attributed to the connectivity of the application and its strength in terms of available APIs for input sources. Specifically, we will be using APIs for both weather and calendar inputs to the application backend. Another strength is to implement a hardware display connected to the application for easier interactions. We have thought of a few possible designs for the mobile part and for the hardware specific part. For the in-closet hardware we have thought about: using LEDs on hangers to display which clothes should be worn on a certain day, making a “smart mirror” which is a mirror with a screen behind it that will display live updates from our application, and using a stationary tablet that would be next to the clothing and receive constant updates and report findings.

Some foreseen weaknesses and trade-offs of the technology is the complications from adding connected devices, such as an mobile application linked to a clothing visual representation in one’s closet. Implementing a strong IOT network will be necessary for this project’s success.

Putting LEDs on the hangers has a few notable weaknesses. The main weakness would be the user. We would have to trust the user to put the same items of clothing back on the same hangers. Not many users would be willing to take even more time when doing laundry in order to ensure the success of an application like this so we ruled this option out.

The smart mirror has a few strengths and weaknesses. Everyone has a mirror in their closet or bathroom so this would just be an augment of a pre-existing household item. If the mirror were to be placed in a bathroom, our electronics would have to be waterproof in order to remain safe. This was a big concern of ours, as well as the fact that the biggest downside would be the time making the electronics would take away from the necessary development time we have in order to make the application work to the best of our ability.

We decided to go with the stationary tablet option. The biggest strength is that we are using pre-existing hardware and can focus on the software part of our project. The application can have a “tablet mode” and a smartphone mode. This would allow us to release only one version of the application but still have it be useful for both the stationary in-closet version and the on-the-go mobile phone version of the application. All of our data will be synced over any devices the user connects to their account and we will not have to worry about there being a difference between the clothing attached to the hardware and the clothing in their database.

2.7 Task Approach

We will be using the Agile methodology for our team. Every two weeks we will evaluate what is the most blocking part of the project to work on at that time and assign programming stories to each team member. Each story will have one name attached to it but multiple people are expected to pair program on each task. The pair programming will ensure that no one member will be solely responsible for a single part of the project and we will all learn as much as we can about the entire technology stack our project uses.

The first screen seen upon entry to the app is a login, which is used for security and account management. This will work with Google’s login service, and is paired with our Firebase backend.

In collecting, we will need to be capable of receiving weather and schedule information. Both of these will be taken care of with respective API calls, with DarkSky as the weather API and Google’s calendar API for schedule details.

The main focus of the app is of course the suggestion for what clothing to wear. This will be completed by taking the calendar and weather information, and checking against what clothing is still in the database as being clean. This should be relatively easy, as we just need to compare what weather specific clothing is acceptable for, as well as whether or not it is available.

Trust the process

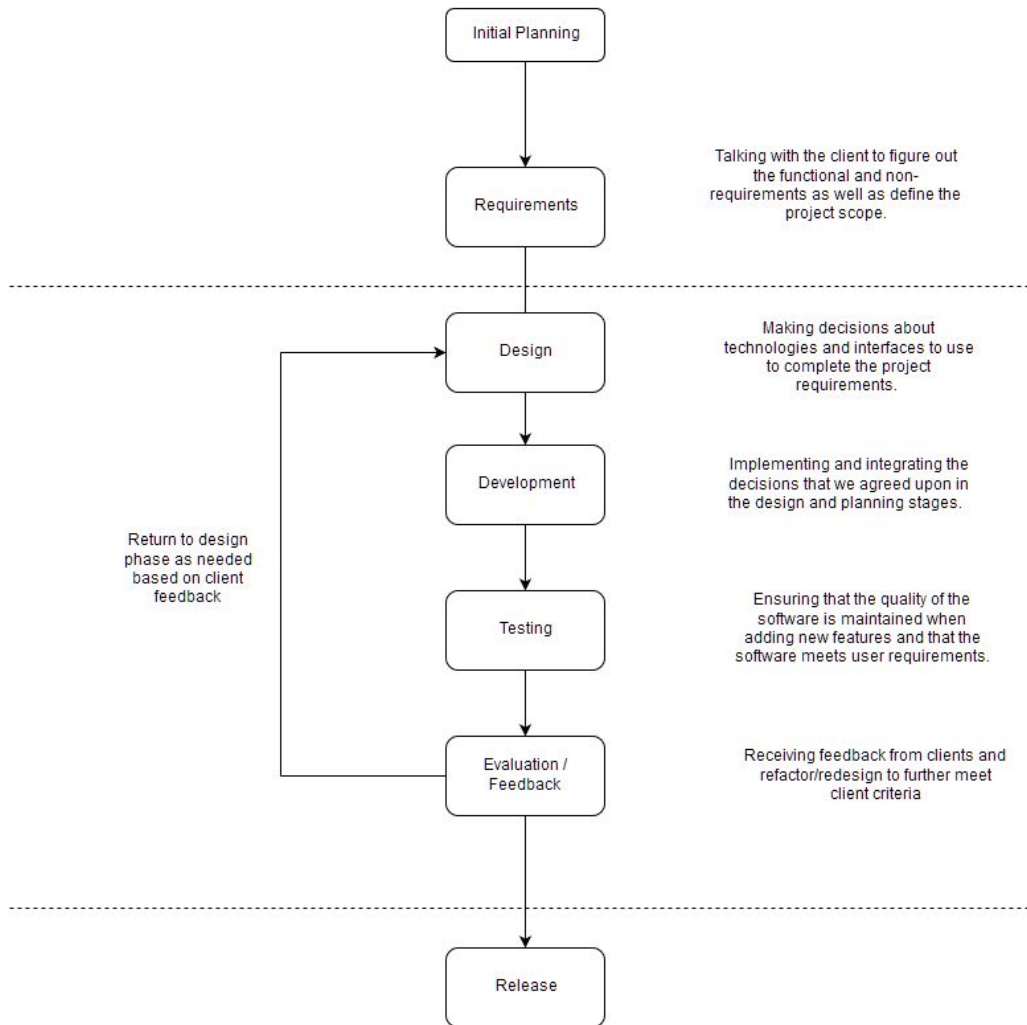


Figure 1: Task Approach Breakdown

2.8 Possible Risks And Risk Management

One of the areas of knowledge we do not currently understand and will be learning as a group is the firebase API service. The initial cost of the firebase license will slow us down a little bit at first, but once we have the license the biggest risk will be everyone learning a new API framework. Only two of us have ever written ReactJS code either and we will have to work as a group to help the others catch up to our skill level and help us all become better ReactJS programmers.

2.9 Project Proposed Milestones and Evaluation Criteria

Phase 1

- Planning

- Confirm project scope
- Architecture Design / Tech Stack Decision

Evaluation Criteria: Once documentation is completed for the tech stack decision and the client informs that our scope and requirements are correct, we will have completed this phase.

Phase 2

- UI Creation
 - UI / Wireframe created
 - Mobile UI created including splash screen
 - Home page with weather display is available
 - Data entry page where users can enter their clothing is available
 - Model to view clothing suggestions is available
- User Login and Storage
 - Registration/Login
 - User is able to log in through the google account login service
 - Store users securely
 - Keeps a copy of the users google account id in the state
 - Makes a global copy in the redux state of the user's information
 - Uses that account id as a reference to any data that user requests of creates
 - Profile Settings
 - User can update settings for their account that are replicated on the database relative to their account id
 - Data entry
 - The user is able to submit their clothes to the database under a table related to their account id
 - We are able to recall a list of all clothes related to any user's account id
- Clothing prediction
 - Consistently retrieve weather data from mobile app
 - Displayed on the homepage and useable in the API calls for the suggestion algorithm
 - Display outfit based on weather data
 - The user is able to view suggested clothing items based on the current weather
 - The suggested clothing items are items out of the users digital closet
 - The digital closet is any clothing items the user entered into their account
 - Working alpha
 - Everything up until this point has been tested and is in working order so that we are able to demonstrate the usefulness of the app
- Integration of external devices
 - Efficient clothing inventory

meetings to discuss what needs to be changed in order to make us more efficient and to evaluate our progress on the project.

Individual velocity

Individual velocity will be tracked by taking the average number of story points completed per sprint. This will be used to estimate progress in future sprints and also check to make sure the individual is on par with team progress.

Team velocity

Team velocity will be tracked the same way as individual velocity, but as a team. This will give us an accurate estimation what we can expect to complete in future sprints and if we are on schedule for completing milestones.

Reference to Gantt Chart

All of these techniques will be used with reference to our gantt chart. We can calculate these estimations and see if the estimation puts us on schedule with the gantt chart. This allows us to keep the big picture in mind and to make sure that we are making progress continually throughout the year. If we are not on schedule according to the gantt chart, we can make adjustments as needed.

Gitlab

Gitlab gives a detailed report for analytics and allows us to track how long it takes to go from receiving the story to committing code, to completing the story. This will give us an accurate estimation for completing stories as well as resolving issues. This will give great insight for the details once stories are assigned. The analytic “code” will be a great tool to use for a team as well.

2.12 Expected Results and Validation

Our end goal of this project is to have a fully function mobile/web application where users can plan their wardrobe for the day based on the weather and calendar events for the day or an extended trip. All of this information for the recommended clothing for the day will be presented to the user upon login, and an ability to search for a location where the application will display the clothing recommendations for the searched destination.

Our efforts are quite evenly split with some developers working connecting and integrating the APIs being used to obtain the weather and calendar information, and some working with the backend and database with authentication for the users to access the wardrobe that they upload to the application.

Testing for the project will be done constantly because of the convenience and ease of running React Native code. Scanning a QR code on a mobile device with the application Expo

will load the project code into the phone, where all additions and refactorings will be evaluated. Tests that will have to be ran will be the functionality of the GUI and the connections to the database and external APIs. The GUI will have to tested to ensure that all buttons, menus, and displays are fully responsive and functional. Then the connections will be tested by ensure the data is being properly fetched from the APIs and sent to the database to report the proper clothing based on the day.

2.13 Test Plan

For each stage of development, we will test new implemented features, as well as performing regression testing in order to ensure that the overall quality of the project stays consistent. This will require us to perform the following tests often, in addition to other tests as needed.

1. Application collects weather data.

Test Case:

For this FR, we want ensure that the application correctly retrieves information from the weather reporting service we decide to use.

Test Steps:

- a. Manually collect weather data.
- b. Check weather data collected by app against manually collected data.

Success Criteria:

This test will be considered successful if the data pulled from the weather service manually is the same as the information pulled by the application.

2. Reasonable recommendations are given.

Test Case:

For this FR, we want to ensure that the system can give the needed recommendations and that the recommendations are reasonable and correct.

Test Steps:

- a. Give program sample wardrobe.
- b. Ensure that the system runs.

Success Criteria:

This test will be considered successful if a large selection of test recommendations meet the specifications set by a default user.

3. User data is saved correctly and retrievable.

Test Case:

For this FR, we want to make sure that user information is correctly stored in the database and retrieved from the database.

Test Steps:

- a. Upload user information.
- b. Load and edit user information through the application.

- c. Check that the information was correctly saved.

Success Criteria:

This test will be considered successful if the sample user data that is entered through the application is correctly represented in the database, and the data can be loaded back into the application from the database without issue.

- 4. Clothing is correctly categorized and displayed.

Test Case:

For this FR, we want to ensure that when a user inputs clothing into the system, it is put into the correct place in the databases and can be accessed again.

Test Steps:

- a. Input clothing items through the application.
- b. Check the databases to ensure that the items were categorized correctly.

Success Criteria:

This test will be considered successful if the clothing input by a test user through the application is stored correctly on the database, then reloaded into the application from the database without issue.

- 5. Application runs on Android and iOS.

Test Case:

For this FR, we want to ensure that the application runs on both Android and iOS at comparable qualities.

Test Steps:

- a. Start the application on an Android and iOS device.
- b. Run all previous tests for both systems.

Success Criteria:

The test will be considered successful if the success criteria of all previous tests are met on both platforms.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Project Timeline

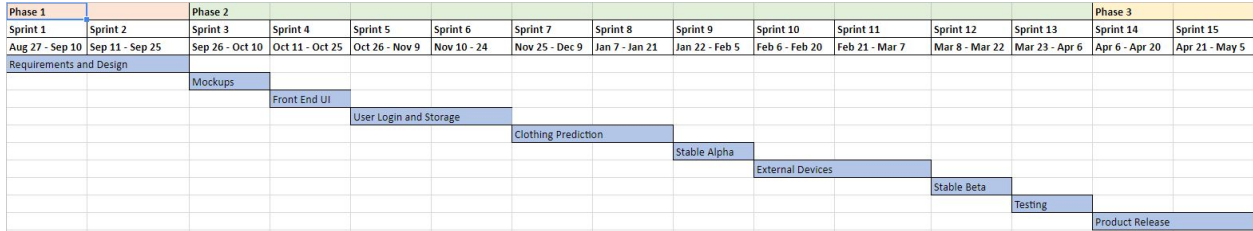


Figure 2: Gantt Chart

As seen above, our team feels that this is the most appropriate timeline given our time constraints. Each sprint represents a two week period from the start of the first semester to the end of the second semester. Roughly each feature should take around two sprints, which allows us to make sure that we have completed the feature given the acceptance criteria and also time to test it in the production environment.

Phase Number	Description	Estimated Time	Comments
1	<p>This phase will be the requirements and design phase. This phase will include determining the scope of the project as well as completing the architectural design of the project.</p> <p>Deliverables:</p> <ul style="list-style-type: none"> • Scope of project • Architecture Diagram of system • Technology stack choices and reports 	2 Sprints	
2	<p>This phase will be the part where there is actual development and will take up the majority of the project. This will include development, wireframes, and further design decisions.</p> <p>Deliverables:</p> <ul style="list-style-type: none"> • Wireframes/Mockups 	11 Sprints	There will be testing done here as well while doing development.

	<ul style="list-style-type: none"> • Alpha version of application • Documentation regarding design choices • Beta version of application 		
3	<p>This phase will be the final transition from beta to the released product. This is where testing will be finalized to ensure a proper release of the product. This will also be the preparation stage to finalize our project as a whole for final presentations. We will also add additional functionality in this phase once the team ensures the product is stable.</p> <p>Deliverables:</p> <ul style="list-style-type: none"> • Testing reports • Final Product and Presentation 	4 Sprints	

Table 1: Timeline Description

As shown in our phase breakdown, we have three different stages. This allows us to see a larger goal when completing the subtasks and what scope we should focus on during that sprint. This allows us to keep the goal in mind and not get sidetracked by non-functional requirements. We felt like the breakdown of initial planning, development, and finalization was the most logical choice because it lines up with the class schedule.

3.2 Feasibility Assessment

After the first major development steps, we will produce a mobile application, which will interface with a weather reporting service, and the user's wardrobe to produce a weather-aware recommendation for the user's clothing. This system will be able to look at weather data in advance in order to let the user plan for trips, and in future iterations, take events in the user's calendar into account when planning wardrobe choices.

We see a small cost estimate for this project. Until this project has been officially released all of the api queries we make will be well within the amount allowed for free each day from the services we are using. We will also require a developers license for the Apple app store in order to post our application in the public domain for iPhone users. We may need to rent an android tablet in order to test out the tablet mode on our application for users to leave on display in their closet.

Foreseen Challenge

The major issues we foresee having for this project:

- Integration issues between multiple services.
- Interfacing with third-party weather app.

3.3 Personnel Effort Requirements

Task	Reference	Estimated development time (hours)	Estimated documentation and testing time (hours)	Total Time (hours)
Create back end API structure	Create the back end API structure for data to be collected from our database table	40	4	44
Create database table for storing user clothes	Create the relational database tables for user clothes lookups	25	5	30
Create database table for storing logins and passwords	Create the relational database tables for username and password hash lookups	30	4	34
Create react front end entry landing page and login page	We will need a login page to determine what user is in the application and what user we will need to look up the clothes for	25	5	30
Create react front end page for displaying what clothes to wear	We will need a front end page that will display the clothing choices made	20	2	22
Create the API lookup for the weather API that will get the data we will test against.	We will need to retrieve the weather data whenever the phone performs a lookup to get the latest weather data from our weather service	20	5	25
Create an "alarm" function that will allow the user to set a daily time they want to be notified of an outfit.	This will be the main chunk of the application what we foresee as the most used function.	30	5	35
Create the ability to send notifications to devices	This will be necessary for any weather updates and for the daily clothes alarm	20	5	25
Create a function to periodically check for weather updates throughout the day and notify the user	This will notify the user any time the weather has significantly changed to necessitate the need for an outfit or attire change such as an umbrella or snow boots.	40	3	40
Create a way to interface with the user's google calendar to get their daily events	This will be to select clothes for the user based on a tiered system of clothes for the day. Business > business casual > casual. The clothing choices for the day will be clothes in the category of the highest tier of clothing for the day based on tags in calendar events.	80	10	90

Create a database table for storing clothing items	This will store the type of clothing, the weather for the clothing, and casualness of the clothing (business, business casual, casual).	20	5	25
Create lookup queries for the database for each of the table	The API queries will be what we call from the react side in order to get the information we need from the query	50	10	60

Table 2: Personnel Effort Breakdown

3.4 Other Resource Requirements

Android and iOS devices to test the application on. A dedicated server to run our back end on.

3.5 Financial Requirements

The costs of this application will depend on the size of its user base. The free version of firebase and of the dark sky API we are using have a query limit per day. After our application goes over that amount of queries there is a small financial cost to make more calls.

We will also require a developer license for Apple in order to post our application on the app store if we take the application to market.

4 Closure Materials

4.1 Conclusion

Our context-aware clothing recommendation system will benefit our users with smart suggestions on what to wear; rain or shine, hot or cold, formal or casual. This will be done by interfacing with the Google location system and an external weather database to get up-to-date weather information in order to give accurate recommendations. The application will also interface with the user's Google Calendar, allowing the user to define events and trips in advance.

We recognize that it can be hard to know what to wear at any given moment throughout the day, whether we are considering weather conditions or specific events on our schedules. That's why we are bringing the best of technology and IOT to bring our users a better way to automatically suggest clothing to wear. We plan to implement an application platform for mobile devices, and a physical representation it's suggestions to point our users to the available clothing options.

4.2 References

List all the sources you used in understanding your project statement, defining your goals and your system design. This report will help you collect all the useful sources together so you can go back and use them when you need them.

- This component shall completely identify any material taken from other sources and used in the development of the project to date or are known that will be used during the remainder of the actual project
- These references shall be complete so that any member of the plan's audience could find them
- Have these on a separate page.

Hussain, Muhammed Mas-ud, et al. "Incorporating Weather Updates for Public Transportation Users of Recommendation Systems - IEEE Conference Publication." *An Introduction to Biometric Recognition - IEEE Journals & Magazine*, ieeexplore.ieee.org/document/7517814.