List of Figures

List of Tables

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

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.

Extending this example would correspond to two kinds of scenarios: (1) a user needs to plan a business trip in multiple places and different geo/climate zones and providing a recommendation for which clothes to pack; (2) an abrupt change in certain parameters (e.g., weather; change of meeting place) necessitates obtaining another clothing item (e.g., a sweater or an umbrella) -- in which case the system should recommend a nearest store or a store requiring smallest deviation from a planned route.

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 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.

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.

2.5 Proposed Design

Design solution 1: Android / iPhone application, app works as an all-in-one solution meant to be displayed on or near a closet space and will provide context-aware clothing recommendations to the user when he/she is nearby the closet space.

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.

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.

2.7 Safety Considerations

None.

2.8 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.

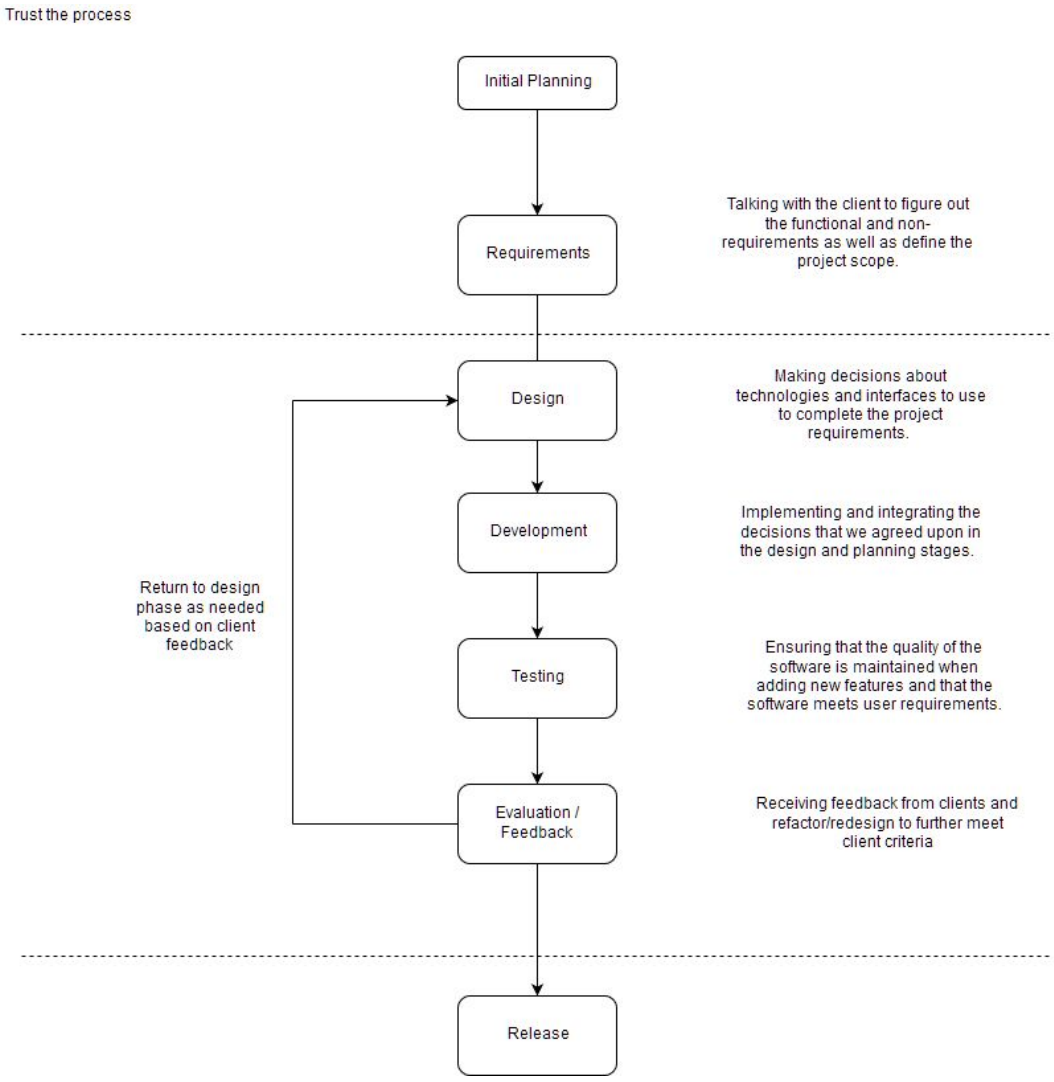

Figure 1: Task Approach Breakdown

2.9 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.

2.10 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
- User Login and Storage
    - Registration/Login
    - Store users securely
    - Profile Settings
- Clothing prediction
    - Consistently retrieve weather data from mobile app
    - Display outfit based on weather data
    - Working alpha
- Integration of external devices
    - Efficient clothing inventory
- Final Touches
    - Push Notifications
    - Working beta

Evaluation Criteria: Once we have completed the features and they are stable in the production environment the feature will be done. Once we have completed all the features, the phase will be considered completed.

Phase 3

- Final Documentation
- Releasing final product

Evaluation Criteria: One the final documentation is accurate then we will release the final product. At that time, we will have completed all the phases.

2.11 Project Tracking Procedures

Trello

To track individual stories, we will use Trello. Trello is a project management software that lets you create a scrum board for tracking stories. Our team will assign stories to individuals and Trello will be used to track what the story is, how difficult the story is, who is working on it. This will keep our team organized and help track progress throughout the semester.

Agile Development

In our project, we will be using the agile development cycle. We will make use of pointing meetings as well as retro meetings to track our progress throughout the year. By using pointing meetings we can see how much we can accomplish during a given sprint. This will be done by calculating velocity for both the team and individuals. Also, we will make use of retrospective 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

What is the desired outcome?
How will you confirm that your solutions work at a **High level**?

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.

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.

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.

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.

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.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Project Timeline

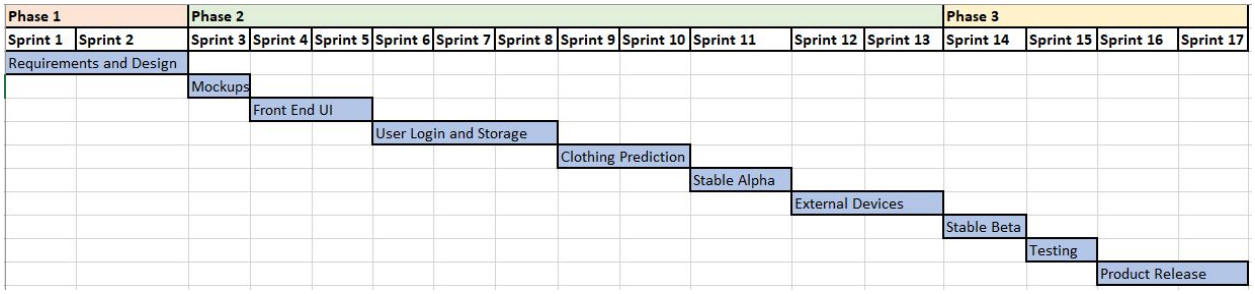

Figure 2: Gantt Chart

As seen above, our team feels that this is the most appropriate timeline given our time constraints. 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 a production environment.

Phase Number	Description	Estimated Time	Comments
1	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. Deliverables: ● Scope of project ● Architecture Diagram of system ● Technology stack choices and reports	2 Sprints	

2	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. Deliverables: ● Wireframes/Mockups ● Alpha version of application ● Documentation regarding design choices ● Beta version of application	11 Sprints	There will be testing done here as well while doing developme nt.
3	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. Deliverables: ● 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.

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 casuality 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

As of now, the project will be entirely contained in a mobile app, but in future, there may be plans to expand the system to encompass a hardware periphery.

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.

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.

4.3 Appendices

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. You may also include your Gantt chart over here.

– Any additional information that would be helpful to the evaluation of the project plan or should be a part of the project record shall be included in the form of appendices

– Examples of project documentation that might be included are property plat layouts or microprocessor specification sheets germane to the proposed project.