CALIFORNIA STATE UNIVERSITY
FULLERTON™

Department of Computer Science

This project has been satisfactorily demonstrated and is of suitable form.

This project report is acceptable in partial completion of the requirements for the Master of Science degree in Software Engineering.

ReMoodWork: A software application for employees managing their mental health during remote work.

Project Title

Peter Ossipov

Student Name

Ning Chen, Ph.D.

Advisor's Name

Advisor's signature Date

Reviewer's Name

Reviewer's signature Date

Abstract: This paper introduces a unique software mental health platform product called ReMoodWork. This software application aims to achieve its goals of aiding employees to manage their mental health in a remote work environment through the current situation with Great Resignation happening primarily in the U.S. and worldwide.

This software application delivers two main characteristics with wide-ranged features and functionalities in developing objectives of working with an employee's mental health tracking through their work culture. Pulse survey will be the first characteristic implementation of a ReMoodWork software platform that works with employees to look further at their daily mental health in remote work. Utilizing enhanced features of a pulse survey will allow employees to enter their work tasks or off-work activity records by writing down their descriptive information and the number of hours spent on a particular project task or break activity used daily for their remote work routine. It will also become a user-friendly and reliable product for employees to check their mental health progress by identifying their work-related questions and emotional statuses of their current pulse survey logs delivered daily to their employers.

ReMoodWork will also utilize a meal planning survey as a second attribute to an employee based on their health, diet, and weight information. This software product achieves in finding a meal between restaurants or recipes that an employer will be responsible for coming up with standard details of a meal item.

In conclusion, ReMoodWork provides a beneficial software application tool for transforming all organizations' remote work environment culture between employees and employers. ReMoodWork will be sure enough to become a potential tool in achieving an employee's daily progress and productivity goal of their mental health with its small steps of creating pulse and meal planning surveys as its primary delivery of the project.

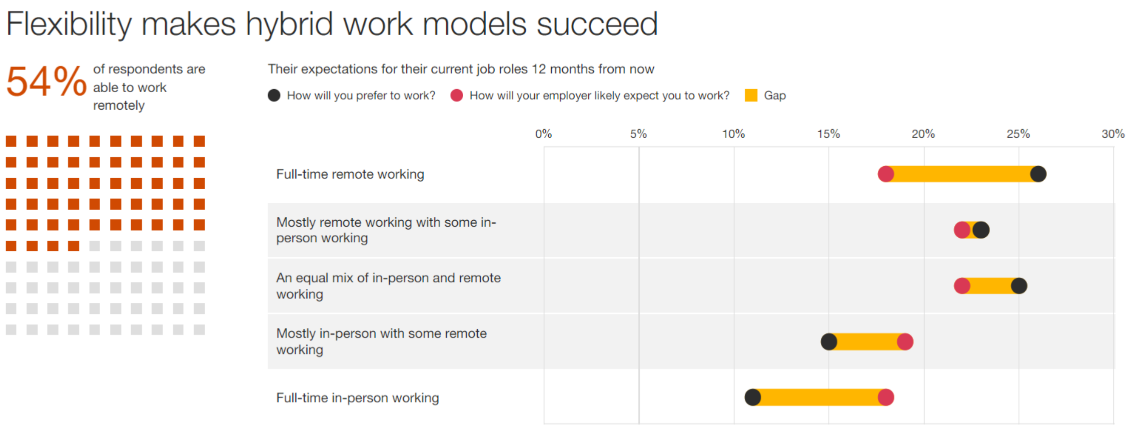
Keywords: *Remote work • Employees • Employers • Pulse Surveys • Meal plan (or assessment) surveys • Django • Model-View-Template (MVT) • Jira Software • Agile • Scrum • Unit testing • Acceptance testing*

Table of Contents

1. Introduction

1. a. Description of the Problem

For quite some time, the Great Resignation has been going on since 2021, with many employees willing to resign jobs. This current trend causes many risks and issues, such as employee dissatisfaction with their full-time job, a standard high cost of living, and a potential fear of COVID-19 as an outbreak. One of the reasons that impacted the beginning of a Great Resignation was for employees to work remotely as their permanent job. According to the figures provided by a forum article, "The Great Resignation is not over: A fifth of workers plan to quit in 2022," found in **Figure 1** of this report, about 26% of employees were satisfied enough to have full-time remote work [1]. Furthermore, 11% stated that employees prefer to work in person as their full-time job [1].



Source: PwC's 2022 Global Workforce Hopes and Fears Survey of 52,195 workers across 44 countries and territories

Figure 1 The Great Resignation is not over: A fifth of workers plan to quit in 2022 [1]

While this report provides a positive future for some employees to live through remote work, it could also negatively impact many employees who have difficulty coping with their mental health as part of their daily routine. Burnout is one of the significant causes that several employees seem to work hard with their mental health problems during remote work, such as stress and anxiety. Studies have shown in the article "Avoiding Stress & Burnout" found on Wrike that about 75% of remote workers have shared their experiences of having to be burnout from their activities [2]. In comparison, 37% have had to work long hours needed to complete their tasks [2].

To its extent, there could be a couple of factors in an attempt to build a software product that could solve all issues of a current remote work environment with the Great Resignation as its primary target. However, this project looks at the two main factors for developing the basis of a unique software application product.

This project first observes its discovery of evaluating and judging employees' daily lifestyle working remotely as its first main issue. How do employees feel when working from home? How many employees have had a burnout in a remote work environment? What are the feeling emotions of an employee working on a task? How many hours and days does an employee spend outside work, such as hanging out with friends and taking small breaks? In response to answering all these questions, this project proposes to utilize a pulse survey as a measuring tool for employee engagement as a potential lead in implementing a software application of this project. Based on some research findings from a few articles online, pulse surveys allow separate companies to measure employees' mental health tracking and experience feedback daily. Furthermore, pulse surveys can also benefit the enhancements of data gathering with the number of meetings occurring per day, time spent outside of work/meetings, and productive result activities made by an employee [3]. This project will also enhance its integrations with emotional pictograms of an employee as Emojis to determine the sense of an emotional behavior for each remote worker in reducing job burnout [4] as part of a pulse survey development feature of a software application project.

This project secondly discusses its message of developing an organized meal plan by creating an assessment survey for employees to fill in their health data information. This project proves that an assessment survey could potentially find the correct information required to find the matched recommended food diet for a meal plan provided to employees related to specific questions. What types of food allergies does an employee have? What are the main goals of measuring employees' health data? Does an employee take their order or make recipes themselves without any takeout? How often does an employee order meal for takeout without cooking it for themselves? These are the possible main questions to develop engaged assessment surveys needed to build parts of a software application system.

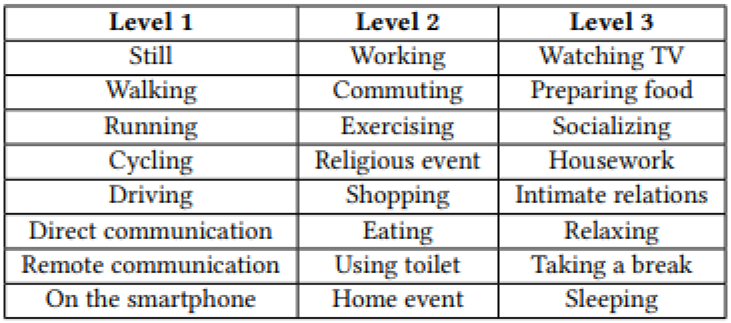
By the end of this project, it will gather a conclusion on how employees manage their mental health and boost company culture through a potentially growing software application designed to improve better reputation and investments of their company. This project shall be able to provide two main attributes needed to deliver its solutions to build a reliable software application system and resolve any problems that a Great Resignation looms over for many employees that this product will offer shown below:

I. A software application shall allow many employers of their company to have an approach for working with their employees on their mental health and daily experiences with pulse surveys and emotional pictograms.

II. A software application shall provide an adapted meal plan to find the perfect food health and diet for employees to choose from the assessment survey. From this project's perspective, employers of their company should offer food services for employees ordering their meals. Employees should be able to select a recipe or ingredients that match their health and diet profile from their assessment survey.

1. b. Project Objectives

In a modern age of technology, it can become much more difficult for employees and employers to maintain specific schedules and conflicts with their feelings and emotional behavior in a remote work environment. Furthermore, these can usually cause a ripple effect through those actions and outcomes of an employee and employer to work remotely, such as feeling sick, having depression, and spending extra hours working on a task. However, this project shall be able to deliver its development process of building a software application for people struggling with mental health. The primary objective of this project is to develop a software recommendation product called ReMoodWork. This application aims at employees to handle their mental health tracking in a remote work environment during the current situations involved with Great Resignation. This software application project will allow employees to keep track of their activity records working remotely. In addition to tracking down an employee's mental health progress, this software application will provide a notification system for employees to sense their emotions with Emojis or emotional pictograms that indicate the status of an employee's mental health condition [4]. The logic of developing work activity logs for a software application product inspires different daily activity levels found in Kahneman's methodology [5] using a few critical levels of emotional pictograms of a given activity.



Level 1	Level 2	Level 3
Still	Working	Watching TV
Walking	Commuting	Preparing food
Running	Exercising	Socializing
Cycling	Religious event	Housework
Driving	Shopping	Intimate relations
Direct communication	Eating	Relaxing
Remote communication	Using toilet	Taking a break
On the smartphone	Home event	Sleeping

Table 1 List of different daily activity levels used from Kahneman's methodology table [5]

The secondary purpose is to develop a meal plan to find measured health and diet information for employees. This project delivers arranged meal plan borrowed from applied food computing to determine food-related data sources and studies [6] to assist employees working in a remote field. This project will provide a unique experience in developing certain specifications and requirements for building a meal plan functionality of a software application. Likewise, this feature will allow employees to measure their stored data on their health, weight, and calorie diet in translating certain suggested meals. From the perspectives of this project, an assessment survey shall interpret its structure as a food log record for employees to track their healthy diet and measure their goals in remote work. In addition to developing its software application, it shall allow an employee to select an ordered meal or request a food recipe through its interpreted food images, a summary description, and calorie information about their meal choice [6].

1. c. Development Environment (software and hardware)

This project provides the development of a software and hardware specifications to run and operate the server of a website application shown below:

Software:

- Programming Languages: Python 3.10.6
- Web Development Framework: Django 4.1.5
- Database Engine: SQLite
- Frontend templates/frameworks: HTML/CSS

Hardware (*This specification was used and operated by an Ubuntu Linux System, although this software application program could work well enough on the Windows 10 as an Operating System environment*):

- Operating System: Ubuntu 22.04.1 LTS for its Linux environments
- Processor: AMD® Ryzen 5 4500u with Radeon graphics × 2
- Virtualization: Oracle VirtualBox
- Memory size of a ReMoodWork directory and its content file name: about 860 KB
- Display Size Resolution: 1024x768 (4:3) Aspect Ratio

1. d. Operational environment (software and hardware)

This project provides software and hardware specifications that will run the server website application and its certain functionalities shown below:

Software:

- Programming Languages: **Python 3.10.6**
- Web Development Framework: **Django 4.1.5**
- Database Engine: **SQLite**
- Frontend templates/frameworks: **HTML/CSS**

Hardware (*This specification was used and operated by an Ubuntu Linux System, although this software application program could work well enough on the Windows 10 as an Operating System environment*):

- Operating System: **Ubuntu 22.04.1 LTS for its Linux environments**
- Processor: **AMD® Ryzen 5 4500u with Radeon graphics × 2**
- Virtualization: **Oracle VirtualBox**
- Memory size of a ReMoodWork directory and its content file name: **about 860 KB**
- Display Size Resolution: **1024x768 (4:3) Aspect Ratio**

2. Requirements Description (external functions and interfaces)

2.1 External functions:

1. **Emoji** module library to test and verify the creation of pulse survey models in the workrecords app in ReMoodWork.

2. Contains Django extensions (**django_extensions**) to show a list of URL routes of remoodwork (can be found in settings.py file).

3. Contains Django Crispy Forms (**django-crispy-forms**) as a format requirement in rendering template pages of ReMoodWork (can be found in settings.py file).

4. Contains **Pillow** module library needed for uploading food meal item images for employers used in ReMoodWork (can be found in settings.py file).

2.2 External interfaces:

2.2.1 Hardware Interfaces

The hardware interfaces shall go along in this project with specific requirements shown below:

1. This application shall accept inputted responses from the keyboard and mouse controller.

2. This application shall have an internet network connection and provide standard protocols to access the website contents of ReMoodWork.

3. This system shall provide an advanced network card for high-speed connectivity of ReMoodWork as its website application.

2.2.2 Software Interfaces

ReMoodWork provides many external interfaces associated with its software infrastructure:

1. *Database Interface:* This system shall provide an object-relational mapping (ORM) structure to represent objects translated from databases to programming used for testing and debugging purposes of a software application.

2. *Graphical User Interface:* This system shall operate and run a web server of a software application through Django.

3. *Web Interface:*
    - *a.* This system shall integrate with a few frontend web technologies with HTML and CSS and be compatible with a wide array of existing browsers like Google Chrome and Microsoft Edge.
    - *b.* This system shall be secured through its proper authentication for authorized users to ensure their access and interactions with other web pages of the ReMoodWork website.

2.2.3 User Interfaces

User interfaces shall provide guidelines with specific requirements related to the software interfaces of an application shown below:

1) The user interfaces of this software application shall be able to integrate and execute other interfaces of the program shown below:

    a. Graphical User Interfaces:
    - The system shall allow an application to present user information through its user-friendly environment.
    - The system shall allow users to interact with the system by visually rendering certain graphical elements, such as a navigational bar containing a list of buttons and dropdown menus.

- The system shall be able to adapt a graphical user interface for its design to enhance user experiences and accommodate straightforward guidance for users to achieve their interactive goals through this software application system.

b. Web Interfaces:

- The system shall breakdown and incorporate specifics parts of a ReMoodWork website application
    A. Data entry form: The system shall allow and accommodate users to enter and record their informational survey data using a data entry form.
    B. Navigational Bar and Menu: The system shall deliver users a layout overview and accessibility to most of the system's features by providing a navigational bar and menu that allows users to click on those single buttons. The system shall include different function buttons explicitly for its characteristics and representation between employees and employers.
    C. Login Page: The system shall allow users to fill in their credentials using a login page functionality to access certain pages and forms of the ReMoodWork website.

c. Database Interfaces:

- The system shall allow users to store and manipulate their data account information and record logs of their surveys.
- The system shall allow users to perform standard types of data manipulation features, such as creating and viewing their list of survey record logs on their employee accounts.

2.3 Functional Requirements

Use cases descriptions and their user stories

Use Case:	User Registration
Use Case ID:	UC01
User Story:	As a user, I want to register my account, so that I can input my credential information and set my job classification status types between an Employee and an Employer before logging into the ReMoodWork site.
Actors:	User
Description:	Users must register their accounts and classify themselves as employees or employers.
Trigger:	Users click a "Register" button
Preconditions:	Users should not be able to input their same username and full name if another user has already created those values.
Basic Flow:	1. The user clicks a "Registration" button on the top-right in the navigational bar of ReMoodWork. 2. The system will redirect a user to the registration page of ReMoodWork. 3. The user inputs their username, password, full name, email, and company name. 4. The users will select "Employee" or "Employer" as their job status class level. 5. The user clicks the "Sign up and register now!" button.

Alternate Flow:	4a. In step 4 of this use case basic flow, if a user selects "Employee" as their job status class level. A. The user will then automatically become classified as an Employee. B. The user then proceeds onto step 5 of this use case basic flow. 4b. In step 4 of this use case basic flow, if a user selects "Employer" as their job status class level. A. The user will then automatically become classified as an Employer. B. The user then proceeds onto step 5 of this use case basic flow.
Exception Flows:	3a. In step 3 of this use case basic flow, if a user inputs the same username as before. A. A prompt message will be displayed to the user stating that the user's username already exists in the system. B. The user gets redirected again to the registration form to fill in a different username back in step 2 of this use case basic flow. 3b. In step 3 of this use basic flow, if a user inputs the same full name as before. A. A prompt message will be displayed to the user stating that the user's full name already exists in the system. B. The user gets redirected again to the registration form to fill in a different full name back in step 2 of this use case basic flow.
Postconditions:	1. The system successfully validates and creates a new user. 2. The system shows the user's username stating that it has created successfully. 3. The system redirects users back to the home page of ReMoodWork.

Use Case:	User Login and Logout
Use Case ID:	UC02
User Story:	As a user, I want to log in with my credentials, so that I can access my private home page in ReMoodWork and safely log out of its system.
Actors:	User
Description:	A user must log into the ReMoodWork platform before accessing certain features and functionalities and to safely be logged out of the system.
Trigger:	Users click a "Login" button
Preconditions:	Users must have a registered account first and not already have logged in and authenticated the ReMoodWork system.
Basic Flow:	1. The user clicks on a "Login" button at the top-middle of a navigational bar. 2. A user visits the login page of ReMoodWork. 3. The user inputs their username and password login credentials. 4. The user clicks the "Sign in" submit button 5. The system takes users to their private home page of ReMoodWork.
Alternate Flow:	5a. In step 5 of this use case basic flow, if a user identifies as an employee. A. The system should contain lists of pulse surveys, meal planning assessment surveys, and restaurant names and food recipe records created by the user represented as an Employee. 5b. In step 5 of this use case basic flow, if a user identifies as an employer. A. The system should contain a list of registered employees associated

	with the same company name and provide a list of food items created as a user represented as an Employer.
Exception Flows:	4a. In step 4 of this use case basic flow, if a user logs in with an incorrect username or password. A. The system will then prompt an error message to the user stating that they must enter the correct username and password. B. The system redirects users back to the login page to fulfill their correct username and password.
Postconditions:	1. The user can successfully log in and access their private home page of ReMoodWork. 2. The user has the option to log out of ReMoodWork. A. If a user logs out of ReMoodWork, the system will redirect them to the logout page of the website.

Use Case:	Create Pulse Survey Record
Use Case ID:	UC03
User Story:	As an employee (user), I want to create a new pulse survey record, so that I can keep track of the daily lifestyle activities of my remote work conditions and notify my employers every day.
Actors:	Employee (Users)
Description:	Employees can create their pulse survey records to stay updated with their employers.
Trigger:	An Employee attempts to create a new pulse survey record.
Preconditions:	Users must first register as an employee and have authorization into their system. The employee should have access to their records list between pulse surveys and meal planning assessments. The employee should have a unique activity name or task not added to the system.
Basic Flow:	1. The user signs in first as a registered Employee. 2. The system takes employees directly to their private home page of ReMoodWork, containing their list of pulse surveys and meal planning assessments. 3. The employee clicks on the "Surveys" button. 4. The employee then clicks the "Visit your pulse survey records" button. 5. The system directs the employee to their pulse survey records page containing their list of pulse surveys. 6. The employee sees "My Pulse Survey Records" on its page and clicks on that button. 7. The employee then clicks on the "Create a new pulse survey record" button 8. The employee inputs their activity name, the number of hours spent on the activity, and any descriptions they want to include related to their current activities or tasks. 9. The employee must choose the type of activity they are doing in their remote work environment. 10. The employee must also choose their emotional rating scale depicted as Emojis and provide their work-related stress issues in their activity.

	11. The employee will then click on the "Submit your pulse survey record" as their new record.
Alternate Flow:	There are no alternative flows to this use case.
Exception Flows:	8a. In step 8 of this use case basic flow, if an employee inputs the same activity name or task. A. The system will display a message to the employee stating that the activity name or task has already existed from their previous pulse survey records. B. The system then redirects the same page to an employee again for creating their new pulse survey records to fill in a different activity name or task back in step 7 of this use case basic flow.
Postconditions:	1. The system will send the employee a prompt message stating that their work tasks or activities have successfully created a new pulse survey record. 2. The system then redirects employees to their pulse survey page to see their new pulse survey records.

Use Case:	Add an employee
Use Case ID:	UC04
User Story:	As an employer, I want to add an employee from my company name, so that I can access their list of survey records that employees have created already.
Actors:	Employer (Users)
Description:	Employers can add new employees from their company name and access their survey records with pulse and meal planning assessment surveys.
Trigger:	An Employer attempts to click an employee name that associates with their company name.
Preconditions:	Users must first register as an employer and have authorization into their system. The employer should have access to a list of employees from their current company.
Basic Flow:	1. The user signs first as a registered Employer. 2. The system takes employers directly to their private home page of ReMoodwork, containing a list of employees associated with their current working company. 3. The employer clicks on the employee's name. 4. The system directs employers to the page where they can add a new employee as their recommended choice. 5. The employer can either click a "Yes" or "No" button.
Alternate Flow:	5a. In step 5 of this use case basic flow, if an employer chooses to select "Yes". A. The system will automatically add an employee to the employer's list. 5b. In step 5 of this use case basic flow, if an employer chooses to select "No". A. The system will redirect employers back to their home page.
Exception Flows:	There are no exception flows needed for this use case.
Postconditions:	1. The system will successfully add a new employee to the employer's

	references. 2. The system will redirect employers to an employee's survey records to see employee's pulse and meal planning assessment surveys.

Use Case:	Create a Meal Plan Record
Use Case ID:	UC05
User Story:	As an employee, I want to add a meal plan, so that I can input my health and dietry contents before selecting food items based on my personal information of my data.
Actors:	Employee (Users)
Description:	Employees can create their dietary information through meal planning assessment record.
Trigger:	An Employee attempts to create a meal planning assessment record.
Preconditions:	Users must first register as an employee and have authorization into their system. The employee should have access to their records list between pulse surveys and meal planning assessments.
Basic Flow:	1. The user signs in first as a registered Employee. 2. The system takes employees directly to their private home page of ReMoodWork, containing their list of pulse surveys and meal planning assessments. 3. The employee clicks on the "Surveys" button. 4. The employee then clicks the "Visit your assessment survey records" button. 5. The system directs the employee to their meal planning record page. 6. The employee sees "Create your meal planning assessment record" on its page and clicks on that button. 7. The employee visits the page to create their meal plan assessment record. 8. The employee inputs their calorie, dietary restrictions, goals, food allergies, budget, and cuisine. 9. The employee will then click on the "Submit your pulse survey record" as their new record. 10. Employees should be able to see their single meal plan record on their meal plan assessment page.
Alternate Flow:	There are no alternative flows to this use case.
Exception Flows:	8. In step 8 of this use case flow, if an employee inputs a negative value number of their calorie information 1. The system will then display a message to an employee stating that their calorie information must be greater than or equal to 0. 2. The system then redirects the same page to an employee again for creating a meal planning assessment record to fill in a positive number of calorie diet information back in step 7 of this use case basic flow.
Postconditions:	1. The system will send the employee a prompt message stating that their meal plan activities have successfully created. 2. The system would then redirect employees to their meal planning page to see their new meal planning assessment records.

Use Case:	Create a Food Item
Use Case ID:	UC06
User Story:	As a employer, I want to add a food item, so that I can include meals made for the employee to select either from a restaurant name or food recipe of their choice.
Actors:	Employer (Users)
Description:	Employers can create a food item based on a restaurant or online recipe.
Trigger:	An Employer attempts to create a new food item
Preconditions:	Users must first register as an employer and have an authorization into their system.
Basic Flow:	1. The user signs in first as a registered Employer. 2. The system takes employers directly to their private home page of ReMoodWork, containing a list of employees and food items. 3. The employer clicks on “Food Records”. 4. The employer then selects “Create a Food Item”. 5. The employer visits the page to create a new food meal item. 6. The employer inputs their food name, description, price, cuisine type, food item type, recipe URL, restaurant name, calories, dietary restrictions, allergy. 7. The employer can optionally upload a food item image if needed. 8. The employer will then click on the “Submit your food item submission” as a new food item. 9. Employers should be able to see their new food meal item on their home page of ReMoodWork.
Alternate Flow:	There are no alternative flows to this use case.
Exception Flows:	6. In step 6 of this use case flow, if an employer inputs a negative value number of their calorie information 1. The system will then display the message to an employer stating that their calorie information must be greater than or equal to 0. 2. The system redirects the same page to an employer again for creating a new food item to fill in a positive number of calorie diet information back in step 5 of this use case basic flow.
Postconditions:	1. The system will send the employer a prompt message stating that it has created a new food item. 2. The system would then redirect employers back to the home page to see a new food item.

Use Case:	Order a Food Meal Item
Use Case ID:	UC07
User Story:	As an employee, I want to order a food item from the website, so that I can place my meal from a restaurant or recipe based on my meal planning assessment profile.
Actors:	Employees
Description:	Employees can create a food meal item based on health and dietary restrictions and information provided from their meal planning assessment

	record.
Trigger:	An Employee attempts to order a food meal item.
Preconditions:	Users must register as an employee and create their meal planning assessment record to order food items.
Basic Flow:	1. The user signs in first as a registered Employee. 2. The system takes employees directly to their private home page of ReMoodWork, containing their list of pulse surveys and meal planning assessments. 3. The employee clicks on the "Surveys" button. 4. The employee then clicks the “Visit your assessment survey records” button. 5. The system directs the employee to their meal planning assessment page. 6. The employee sees “My Meal Plan Records” and “Meal choices that are recommended for you” on the page. 7. The employee sees a food meal item (Ex. Smoked Salmon) and clicks on the “Select” button. 8. The system directs employees to the page where they can choose to order a food item of their choice. 9. The employer can either click a “Yes” or “No” button.
Alternate Flow:	9a. In step 9 of this use case flow, if an employee chooses to select “Yes”. A. The system will automatically place an ordered food meal item from an employee’s decision. B. The system prompts a successful message to the user stating “You have successfully placed an ordered” of a food meal item from an employee’s selection. 9b. In step 9 of this use case flow, if an employee chooses to select “No”. A. The system will redirect employees back to their meal planning page and has a choice to choose the same or different food meal item.
Exception Flows:	There are no exception flows needed for this use case.
Postconditions:	1. The system will successfully place a new food meal order for the employee. 2. The system will redirect employees back to their meal planning assessment and will not see a “Select” button for the food item they ordered.

2.4 Nonfunctional Requirements

2.4.1 Usability

A. US-1: The software application product shall become intended to prevent user errors and confusion and manage a clear layout of its labeling and specific instructions.

B. US-2: The software application product shall have sensitive user-friendly environments delivered for users working remotely that require minimal knowledge and training of the complex infrastructure of its creation made for the users.

2.4.2 Portability

A. PT-1: The software application shall become compatible and established with other database management systems.
B. PT-2: The software application shall allow users to access ReMoodWork using multiple browsers like Google Chrome, Microsoft Edge, and Mozilla Firefox.
C. PT-3: The software application shall integrate from its website into a mobile application for users, such as iPhone and Android smartphones.

2.4.3 Performance

A. PR-1: The software application shall provide its availability of at least 99% scheduled uptime and downtime of its systems.
B. PR-2: The software application shall request throughput data to process about 1500 transactions per minute during high peak hours.
C. PR-3: The software application shall fully render its webpages at a 5 seconds average or less from an internet connection.

2.4.4 Security

C. SEC-1: The software application shall be able to detect and monitor security incidents and vulnerabilities happening from the user.

3. Design Description (architecture, internal functions and interfaces)

3.1 ReMoodWork's Designed Architecture Overview

The descriptions of this software design summarize the architectural infrastructures of the ReMoodWork application platform. Furthermore, this software system is solely built and implemented on a Model-View-Template (MVT) as an architectural software design pattern to deliver a simple, organized way of arranging unique source codes and its data presentation, logic, and model for producing an overall reliable quality of its mental health application for remote workers.
The software architecture's structural data flow pattern depicts itself in **Figure 2** of this Report and provides its steps used in **Figure 2** shown below:

1. The user sends their data input requests to the ReMoodWork website application.
2. The website then sends the user's request to the Django server based on a few application names of ReMoodWork shown below:
   a. Users
   b. Work Records
   c. Meal Plans
3. The server sends the requests to a selected URL dispatcher route (**urls.py**) that the user is requesting from the ReMoodWork website.
4. The URL route sends the user's request to the views component of the Model-View-Template (MVT) architecture used for ReMoodWork.
   a. **views.py**: It controls the logic mechanics and communications between models, forms, and templates from the user's request.
5. The views then send the user's request to the forms (**forms.py**) component responsible for saving and creating data information before sending the response message back to the users that use the ReMoodWork website.

6. Once the forms are saved and stored in the database models (**models.py**) of ReMoodWork, it will begin to process and send its message response and render the template (**templates/app_name**) webpages of ReMoodWork back to users.

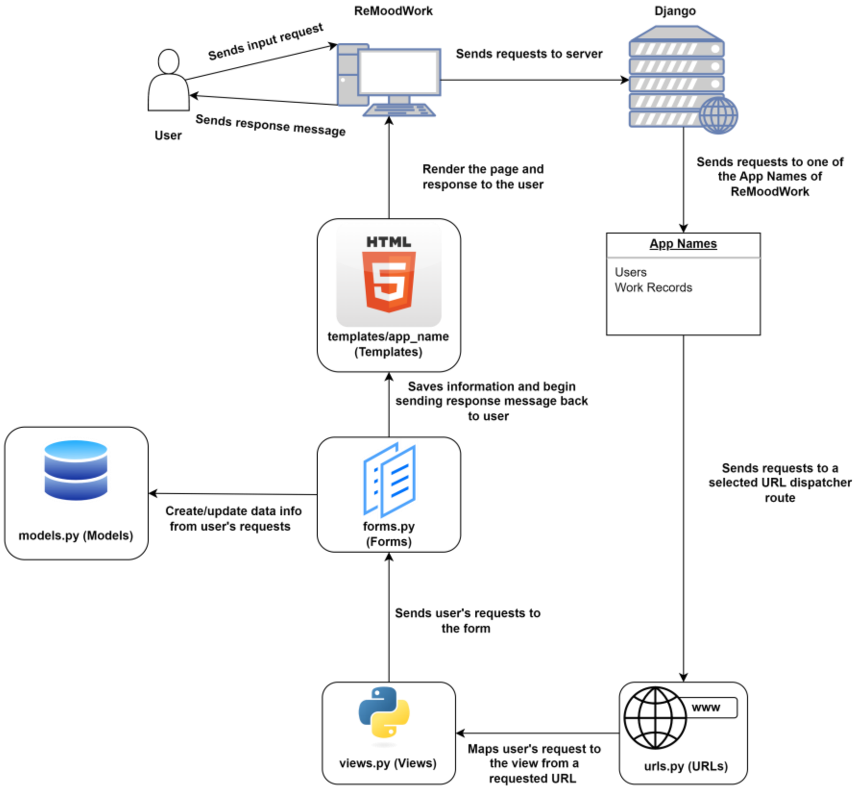

ReMoodWork
Django
Sends input request
Sends requests to server
Sends response message
User
Render the page and response to the user
Sends requests to one of the App Names of ReMoodWork
HTML
templates/app_name (Templates)
App Names
Users
Work Records
Saves information and begin sending response message back to user
models.py (Models)
Create/update data info from user's requests
forms.py (Forms)
Sends requests to a selected URL dispatcher route
Sends user's requests to the form
views.py (Views)
Maps user's request to the view from a requested URL
www
urls.py (URLs)

Figure 2 Structural data flow pattern of ReMoodWork software architecture via Model-View-Template (MVT)

3.2 ReMoodwork's Internal Functions and Interfaces

An architectural design pattern of a Model-View-Template (MVT) has become deliberately made for the internal functions and interfaces of ReMoodWork's software application using the three main components needed for its platform shown below:

- Models (**models.py**): Represents itself as an internal interface for depicting specified data models of ReMoodWork's system, such as users, employees, employers, pulse surveys, and assessment meal planning surveys used as the primary models of this development project. These data models represent the structures and relationships stored in an application's database management system.
- Views (**views.py**): Represents itself as an internal interface that only contains built-in internal functions needed for this development project of ReMoodWork. It is solely responsible for operating and sending direct responses based on the user's requests for notifying with templates to render HTML webpages and models to read and control data information from the database management system.
- Templates (**templates/app_name**): Represents itself as an internal interface to provide an overview layout structure of all displayed HTML webpages presented to the users. Templates are also responsible for giving fetched data between views and models to deliver user experiences visually for the usability of ReMoodWork's software application.

Furthermore, the URL endpoints and Forms also contain their internal functions and interfaces used for integrating with Model-View-Template as part of an architectural design pattern for handling user interfaces and inputs used through the software application of ReMoodWork shown below:

- URLs (**urls.py**) represent the internal interface for mapping out specific URL routing patterns and are responsible for handling certain requests appropriate to the internal functions of views that are part of a Model-View-Template (MVT) architectural design for the software application website of ReMoodWork.
- Forms (**forms.py**) also represent the internal interface for gathering inputs made by the client and are sent for processing with the server requests with data security and field validation to integrate with a Model-View-Template (MVT) architectural design pattern for the software application website of ReMoodWork.

The characteristics of ReMoodWork apply to several internal functions and interfaces of a Model-View-Template (MVT) by describing all parts for each application names of its development project shown below:

3.2.1 Users Application

The responsibilities of implementing the users' application of ReMoodWork are to work with users to register and provide authentication for users to log in with their accounts by using the models, views, templates, URLs, and forms shown in the following subsection headings of the users' application name. It also covers a feature for employers to add an employee based on the same company name.

3.2.1.1 Models (Internal Interface):

The users' application provides the following data models shown below:

- **User:** This model extends with Django's built-in AbstractUser model to provide username and password fields inherited from another Django's built-in model: User [7]. It also includes additional fields in the users' models.py file, such as full name, email, company name, and job classification, for users to select either an employee or an employer.
- **Employee:** This model provides a one-to-one relationship with the users' model, meaning that each user can only have one profile represented as an employee of their company name.
- **Employer:** This model provides a one-to-one relationship with the user's model, meaning that users can only have one profile represented as an employer of their company name. It also delivers a many-to-many relationship allowing each employer to become associated with one or more employees associated with the same company name.

3.2.1.2 Views (Internal Interface and Functions):

The users' application contains built-in views (internal interface) represented as internal functions for handling user registration and authentication process of the ReMoodWork application. It also includes some internal functions to control certain employees' and employers' actions of the ReMoodWork application shown below:

- **Register View:** This view executes user registration by presenting a built-in registration form for the user to submit and save their data attribute information and their job classification status of either an employee or an employer stored in the database. It is also responsible for validating user inputs to verify the completed, submitted data used in ReMoodWork's system.
- **Login View:** This view controls the users' login information by presenting a login form to the user and ensuring that authenticated users can access the private pages of ReMoodWork after

submitting their form to the system. The user can become redirected to their private home page of ReMoodWork if they are authenticated successfully from the login page. This view could also become likely to prompt an error message to the user if their authentication process fails on the system's login page.

- **Logout View:** This view is only accessible to authenticated users logged into the system from the login page. Furthermore, all authenticated users can become logged out and potentially fulfill their login credentials again from the login view controller of the ReMoodWork website system.
- **Add an Employee View:** This view is only accessible by an employer to decide on selecting and adding an employee under those following conditions shown below:
    - If selected "Yes," the system will append an employee to the employer's employee list record.
    - If selected "No," the system will redirect an employer back to their private home page of ReMoodWork to choose the same or different employee.

3.2.1.3 Templates (Internal Interface):

Templates are in charge of rendering HTML webpages within the user's application and are known for displaying and accommodating user-oriented pages of the ReMoodWork application shown below:

- **register_page.html:** This template displays and renders data information from the user's registration form. It contains fields to fulfill users' information using their username, password, full name, email address, and company name. Additionally, it provides a job classification choice required for users to select an employee or employer in the user registration form. It also determines a logic validation process to confirm that the user's specific input fields are correct, such as username and full name validation.
- **login_page.html:** This template displays and renders data information from the user's login form. It must require a username and a password field to allow users to log in and access the personal pages of ReMoodWork. Before accessing certain pages of ReMoodWork, this template can also determine a logic validation process to confirm that the username and password from the user are the correct input values before successfully becoming authenticated through its website application.
- **logout_page.html:** This template displays a simple, user-friendly message of the webpage for users stating that they have successfully been logged out of the website so that they can log in directly back to the public home page of ReMoodWork.
- **add_an_employee_page.html:** This template displays a page for an employer to decide and add an employee of the same company name.

3.2.1.4 URLs (Internal Interface):

The users' application provides several URL routing endpoint names shown below:

- **register/ ("remoodwork-register-user"):** This URL route takes users directly to the registration page to input their credentials before accessing several webpages of ReMoodWork.
- **login/ ("remoodwork-login-user"):** This URL route takes users directly to the login page of ReMoodWork, where they will input their username and password credentials to process their valid authentication through the software application website of ReMoodWork.
- **logout/ ("remoodwork-logout-user"):** This URL route takes directly to the logout page for authenticated users within the software application website of the ReMoodWork.
- **employer/<int:pk>/add_employee/employee/<int:emp_pk>/ ("remoodwork-add-employee"):** This URL route takes employers to the page to decide whether they want to select an employee from the same company name.

3.2.1.5 Forms (Internal Interface):

The users' application provides its defined built-in forms to handle user registration and authentication management process saved in a software application website of ReMoodWork. It also provides a few internal function methods and other built-in internal interface forms from Django used for controlling the user management process through the list of form class names shown below:

- **UserRegistrationForm:** Used for defining the six data attributes fields for users required to register their credential information made for the software application system of ReMoodWork, such as username, password, full name, email address, the current company name, and job classification choice between an employee or an employer of their company name. This class inherits from the UserCreationForm class as one of the internal interfaces of the Django forms required for users to enter their password twice (like password1 and password2). This class also provides a few internal function methods shown below:
    - ***save(commit=True):*** This internal function method is used to save all the six data attribute fields of the users and returns the instance of a new user created and recorded through the database records of the ReMoodWork software application. It also creates a new employee or employer based on the job classification of their choice saved in the UserRegistrationForm class of ReMoodWork.
    - ***is_valid():*** This internal function method is used to verify and check if a user's username and full name already exist through the database record models of the ReMoodWork software application based on the inputs provided by the user using those two conditions shown here:
        - ❖ If a username and full name do not exist from the database records of ReMoodWork, then this method will return true.
        - ❖ If either a username or full name exists from the database records of ReMood, then this method will return false and include an error message made for either a username or the user's full name.
- **UserLogInForm:** Used for handling user authentication methods to only include username and password as the primary attribute fields required for logging into the software application system of ReMoodWork. There are no internal function methods used in this class. However, this class only inherits from the AuthenticationForm class as one of the internal interfaces of the Django forms to handle the authentication process fulfilled from the user's credential information.

3.2.2 Work Records Application

The responsibilities of implementing the work records application of ReMoodWork are to handle and track down several work record logs using pulse surveys and meal planning assessment surveys created by an employee. Furthermore, it will monitor survey records of an employee by using the models, views, templates, URLs, and forms shown in the following subsection headings of the work records application name.

3.2.2.1 Models (Internal Interface):

The work record's application provides the following data models shown below:

- **Pulse Survey:** This model represents a pulse survey created and managed by employees to be used and monitored by their employers of the same company name. The data attribute fields of a pulse survey contain the title of a work activity name, the date of an activity, the number of spending hours, and descriptions of a work activity name. This model provides a one-to-many relationship to an employee, where a single employee can request to create several pulse survey records made on a ReMoodWork platform application. It also includes three choice fields for creating and monitoring pulse survey records shown below:

  - The first choice data attribute, named activity type, represents different types of work activities and tasks using those three indicator names "Work task," "Take a break," and a "Day off/vacation" provided by an employee to work remotely.
  - The second choice data attribute, emotional rate status, represents the emotional behaviors of an employee working remotely by utilizing certain Emoji pictogram faces, such as a happy, sad, or tired face, as leading examples for the pulse survey model record.
  - The third choice data attribute, work stressor status, provides information to employees stating if they have had any stress-related work activity or task issues in their remote work environment.

- **Meal Plan:** This model represents a meal plan assessment created and managed by employees to be used as their profile to provide dietary health information. The data attribute fields of a meal plan containing calories, dietary restrictions, goals, allergies, food budget, and cuisine for an employee. This model provides a one-to-one relationship to an employee, where an employee makes a one-time providing their input informational field of their meal planned assessment records made on a ReMoodWork platform application.
- **Food Item:** This model represents a food item product produced by an employer. The data attribute fields of a food item contain the title of the food name, food description, price, recipe URL (link), restaurant name, and calories. This model also allows employers to upload an image of a food item. This model provides a one-to-many relationship to an employer, where a single employer can create multiple food items made in ReMoodWork’s platform application. It also includes four choice fields for producing food items shown below:
  - The first choice data attribute, name food item type, represents a meal choice for employees to select using “Restaurant” and “Recipe” as two of the options needed to determine the categorization of a food type item.
  - The second choice data attribute, cuisine types, represents a cooking meal variety based on distinguished styled dishes for a food product item that only contains “Italian,” “American,” “Mexican,” and “Asian” as examples of choices used in this project.
  - The third choice data attribute dietary restrictions, allows employers to select a dietary indication based on the food item meal with “Gluten-free” and “Vegetarian,” as examples used in this project.
  - The fourth choice, data attribute allergy, allows employers to select allergy types to indicate and determine food meal items made for employees, using “Wheat,” “Peanuts,” “Shellfish,” and “Beef,” and “Soy” as allergy examples prompted in this project.
- **Order:** This model represents a meal order to place and save food items made by an employee. This model provides a one-to-many relationship to an employee and their optional meal plan. A single employee can create multiple meal orders to select all the food items in a many-to-many relationship presented in ReMoodWork’s platform application.

3.2.2.2 Views (Internal Interface and Functions):

The work record's application contains built-in views (internal interface) represented as internal functions for handling and managing survey records used by several employees for the ReMoodWork application shown below:

- **Home Page:** This view executes logical information with the server and client side of the main home page for ReMoodWork. Furthermore, this view utilizes itself to control and display distinguished data information made for employees and their employers in this software application. This view should provide valuable information for employees to show their list of pulse surveys and meal planning assessment surveys used on the main website home page of ReMoodWork. This view should also deliver helpful information to their employers on retrieving a list of employee names filtered by the same company names associated with the targeted employer and provide a list of food items created on the main website home page of ReMoodWork.
- **Pulse Survey View:** This view executes a list of pulse survey log records created by a given employee. Furthermore, this view is responsible for requesting and fetching contents of an employee's pulse survey borrowed from the database and presenting the source data information displayed to the user for its user-friendly environment of ReMoodWork.
- **Create a Pulse Survey View:** This view allows employees to create their pulse survey records by presenting a built-in pulse survey creation form for an employee to submit and save data attribute information of their pulse survey records, which also includes their activity types, emotional rate behaviors, and work stressor statuses stored through databases. It is also responsible for validating employee inputs to verify the completed, submitted pulse survey data used in ReMoodWork's system.
- **Meal Plan View:** This view executes a meal plan assessment record created by a given employee. Furthermore, this view is responsible for requesting and fetching contents of an employee's meal plan assessment borrowed from the database and presenting the source data information displayed to the user for its user-friendly environment of ReMoodWork. It also displays a list of food meal items for employees to select.
- **Create a Meal Plan View:** This view allows employees to create their meal plan assessment record by presenting a built-in meal plan form for an employee to submit and save data attribute information of their meal plan record. It is also responsible for validating employee inputs to verify the completed, submitted meal plan data used in ReMoodWork's system.
- **Create a Food Item View:** This view allows employers to create a food meal item by presenting a built-in food item form for an employer to submit and save data and file attribute information of a food meal item. It is also responsible for validating employer inputs to verify a submitted food item used in ReMoodWork's system.
- **Order Food Meal Item View:** This view allows employees to make a decision on placing a meal order for a food item based on the following conditions:
    - If selected "Yes," the system will place an ordered meal item made by an employee.
    - If selected "No," the system will redirect an employee back to the meal planning page to choose a new or different food item.

3.2.2.3 Templates (Internal Interface):

Templates are reliable for rendering HTML webpages within the work records application. They are known for displaying and accommodating user-oriented pages of the survey records made for the ReMoodWork application. It also displays the main home page of ReMoodWork shown below:

- **home_page.html:** This template displays the home page as the central hub area for ReMoodWork's website. This template also renders distinguished results based on the information by an employee and employer from the home page view function shown below:

  - As an employee, It will only display their list of pulse surveys and meal planning assessment surveys provided to employees who keep track of their history of survey records found on the ReMoodWork's home page website.
  - As an employer, It will only display a list of employees associated with the same company name they are currently working and display list of food items created by an employer.
- **base.html:** This template displays the basis of a navigational bar and renders different buttons for several stakeholders between users, employees, and employers to perform certain tasks and features for the website infrastructure of ReMoodWork.
- **pulse_survey_main_page.html:** This template renders a list of pulse survey records created by a given employee that display their work activity names, as well as their type and date of an activity, the number of hours spent on the activity, their activity descriptions, and their emotional rating scale and work stressor statuses as a tracking record for employers to see and review in the future.
- **create_pulse_survey_page.html:** This template displays and renders from a pulse survey creation form. It contains fields needed for employees to create their pulse survey records daily by inputting work activity name, type, and the number of hours spent on the activity for working remotely. It also requires employees to input their emotional rating scale and work stressor statuses to be saved in creating a pulse survey records form. Employees can optionally include their description of an activity name if they want to include that for creating their pulse survey records.
- **meal_plan_main_page.html:** This template renders a meal-planning assessment record created by a given employee that display their calorie information, as well as their dietary restrictions, goals, food allergies, food budget, and cuisine as a tracking record.
- **create_meal_plan_page.html:** This template displays and renders from a meal planning creation form. It contains fields needed for employees to create their meal-planning records by inputting calories, dietary restrictions, goals, food allergies, food budget, and cuisine during remote work.
- **create_food_item_page.html:** This template displays and renders from a food item creation form. It contains fields for employers creating food items by inputting food names, descriptions, prices, recipe URLs (link), restaurant names, calories, and a food meal image. It also requires employers to input their food item types, cuisine types, dietary restriction types, and allergy types to be saved in creating a food item form. Employers can optionally provide a description section of a food name and upload images of a food item. Employers can include a recipe URL if the food item type selects “Recipe.” Employers can also have the restaurant name and price of the food item if chosen as a “Restaurant.”
- **order_food_meal_plan.html:** This template displays and renders a page for employees to decide on placing their meal order of a food item.

3.2.2.4 URLs (Internal Interface):

The work records application provides several URL routing endpoint names made for a couple of stakeholders shown below:

- **("remoodwork-home"):** This URL route takes users, employees, and employers to the main website home page of ReMoodWork.

- **employee/<int:pk>/pulsesurvey/ ("remoodwork-pulse-survey"):** This URL route takes employees to the page containing their list of pulse survey records created for the ReMoodWork website.
- **employee/<int:pk>/pulsesurvey/employer/<int:emp_pk>/ ("remoodwork-pulse-survey"):** This URL route takes employers to the page to see the list of pulse survey records created by their specific employees used for ReMoodWork website.
- **employee/<int:pk>/createpulsesurvey/ ("remoodwork-create-pulse-survey"):** This URL route takes employees to the page for creating their pulse survey records used for the ReMoodWork website.
- **employee/<int:pk>/mealplan/ ("remoodwork-meal-plan"):** This URL route takes employees to the page containing their meal planning assessment record and a list of food meal items for the ReMoodWork website.
- **employee/<int:pk>/createmealplan/ ("remoodwork-create-meal-plan"):** This URL route takes employees to the page for creating their meal plan record used for the ReMoodWork website.
- **employer/<int:emp_pk>/createfooditem/ ("remoodwork-create-food-item"):** This URL route takes employers to the page for creating food meal items used for the ReMoodWork website.
- **employee/<int:pk>/ordermeal/<int:food_pk> ("remoodwork-order-meal"):** This URL route takes employees to the page to place their meal order of a selected food item.

3.2.2.5 Forms (Internal Interface):

The work records application provides built-in forms to allow employees to create their survey log records with pulse surveys and meal planning assessment surveys saved in a software application website of ReMoodWork. It also provides a few internal function methods used for controlling survey record processes through the list of form class names shown below:

- **PulseSurveyCreationForm:** Used for defining five required data attribute fields and one optional data attribute field for users creating their pulse survey records made for the software application system of ReMoodWork, such as work activity name and type, as well as the number of hours spent on the work activity or task, the emotional rating scale, an optional description of the work activity name, and a work stressor status of an activity name. This class provides a few internal function methods shown below:
    - ***is_valid():*** This internal function is used to verify and check if an activity name from a recently created pulse survey already exists through the database record models of the ReMoodWork software application based on the inputs provided by the user using those two conditions shown here:
        - ❖ If an activity name of the pulse survey does not exist from the database records of ReMoodWork, then this method will return true.
        - ❖ If an activity name of the pulse survey exists from the database records of ReMoodWork, then this method will return false and include an error message made for the activity name of the pulse survey records to its employees.
- **MealAssessementCreationForm:** Used for defining six required data attribute fields for users creating their meal plan assessment records made for the software application system of ReMoodWork, such as calories, dietary restrictions, goals, allergies, food budget, and cuisine. This class provides no internal function methods.
- **FoodItemCreationForm:** Used for defining six required data attribute fields and four optional data attribute fields based on determining the food item as a restaurant or recipe used in ReMoodWork. It contains the food name, description, price, cuisine type, food item type, recipe

URL, restaurant name, calories, dietary restrictions, allergy, and an image of a food meal. This class also provides no internal function methods.

4. Implementation (organization of source file structure, reference list of files)

This section provides important directory folders and source code files used in ReMoodWork's codebase structure of this project. Furthermore, this project delivers its implementation through a Model-View-Template architectural pattern. It follows a standard directory structure containing starter files used in ReMoodWork shown below:

```
.
├── .gitignore ← Ignores additional libraries (my-django-env as an example)
├── README.md
├── remoodwork ← ReMoodWork project
│   ├── db.sqlite3 ← SQLite DB (Please see 'Installation Instructions' for migrate command)
│   ├── manage.py ← Used to run management commands and interact with ReMoodWork
│   ├── media
│   │   ├── default.jpg ← default image if an employer is not uploading images to their food item
│   │   └── meal_item_images ← System automatically uploads images to employers creating food items
│   │       ├── Example_Salmon.jpg
│   │       └── Example_Steak.jpg
│   ├── remoodwork ← Contains main settings and configuration files of ReMoodWork
│   │   ├── asgi.py ← ASGI configuration for ReMoodWork
│   │   ├── __init__.py
│   │   ├── settings.py ← Configuration settings of ReMoodWork (database, installed, etc.)
│   │   ├── urls.py ← URL route for admin and all users (login, logout, registration, home page)
│   │   └── wsgi.py ← WSGI configuration for ReMoodWork
│   ├── users ← Users application
│   │   ├── admin.py ← Register Users, Employees, and Employers for admin site of ReMoodWork
│   │   ├── apps.py
│   │   ├── forms.py ← User forms (Registration and Login form)
│   │   ├── __init__.py
│   │   ├── models.py ← User models (User, Employer, Employee)
│   │   ├── templates ← User HTML files to display webpages of ReMoodWork
│   │   │   └── users
│   │   │       ├── add_an_employee_page.html
│   │   │       ├── login_page.html
│   │   │       ├── logout_page.html
│   │   │       └── register_page.html
│   │   ├── tests ← Directory to test all components of Users app (model, form, and view)
│   │   │   ├── __init__.py
│   │   │   ├── test_base.py
│   │   │   ├── test_login_user_form.py
│   │   │   ├── test_user_registration_form.py
│   │   │   ├── test_users_models.py
│   │   │   └── test_users_view.py
│   │   ├── urls.py ← Additional users' URL routes
│   │   └── views.py ← User's view functions and HTTP requests
│   └── workrecords ← Work Records application
│       ├── admin.py ← Register Work Record models for ReMoodWork admin site
│       ├── apps.py
│       ├── forms.py ← Work Record forms (Creating Pulse Survey, Meal Plan, and Food Item)
│       ├── __init__.py
│       ├── models.py ← Work Record models (Pulse Survey, Meal Plan, Food Item, and Order)
│       ├── templates ← Work Record's HTML files to display webpages of ReMoodWork to the user
│       │   └── workrecords
│       │       ├── base.html
│       │       ├── create_food_item_page.html
│       │       ├── create_meal_plan_page.html
│       │       ├── create_pulse_survey_page.html
│       │       ├── home_page.html
│       │       ├── meal_plan_main_page.html
│       │       ├── order_food_meal_plan.html
│       │       └── pulse_survey_main_page.html
│       ├── tests ← Directory to test all components of Work Record’s app (model, form, and view)
│       │   ├── food_meal_images_test ← Contains food images of an item to test/upload by employers
│       │   │   ├── Example_Salmon.jpg
│       │   │   └── Example_Steak.jpg
│       │   ├── __init__.py
│       │   ├── test_base.py
│       │   ├── test_food_item_creation_form.py
│       │   ├── test_food_item_creation_view.py
│       │   ├── test_food_item_model.py
│       │   ├── test_meal_plan_creation_form.py
│       │   ├── test_meal_plan_creation_view.py
│       │   ├── test_meal_plan_model.py
│       │   ├── test_order_food_item_model.py
│       │   ├── test_order_food_item_view.py
│       │   ├── test_pulse_survey_creation_form.py
│       │   ├── test_pulse_survey_creation_view.py
│       │   └── test_pulse_survey_models.py
│       ├── urls.py ← All URL routes of Work Records for the user information
│       └── views.py ← Work Record’s view functions and HTTP requests for user’s information
└── requirements.txt ← Text file to install Django and other modules
```

5. Test and Integration (plan and results)

5.1 Plans for the development project of ReMoodWork

Before describing management plans used for this development project, there were no daily standup team meetings or sprint retrospectives for each team to review and go over the tasks and duties they have worked on this project. However, the plan was arranged and managed by a single person that recorded the release sprint plans for each developed iteration of the entire project using Jira Software as a project management tracking tool. The testing plans of this project were also managed and evaluated by a single person to write specific test cases and obtain the final results of a software application for the verification and validation of its development project.

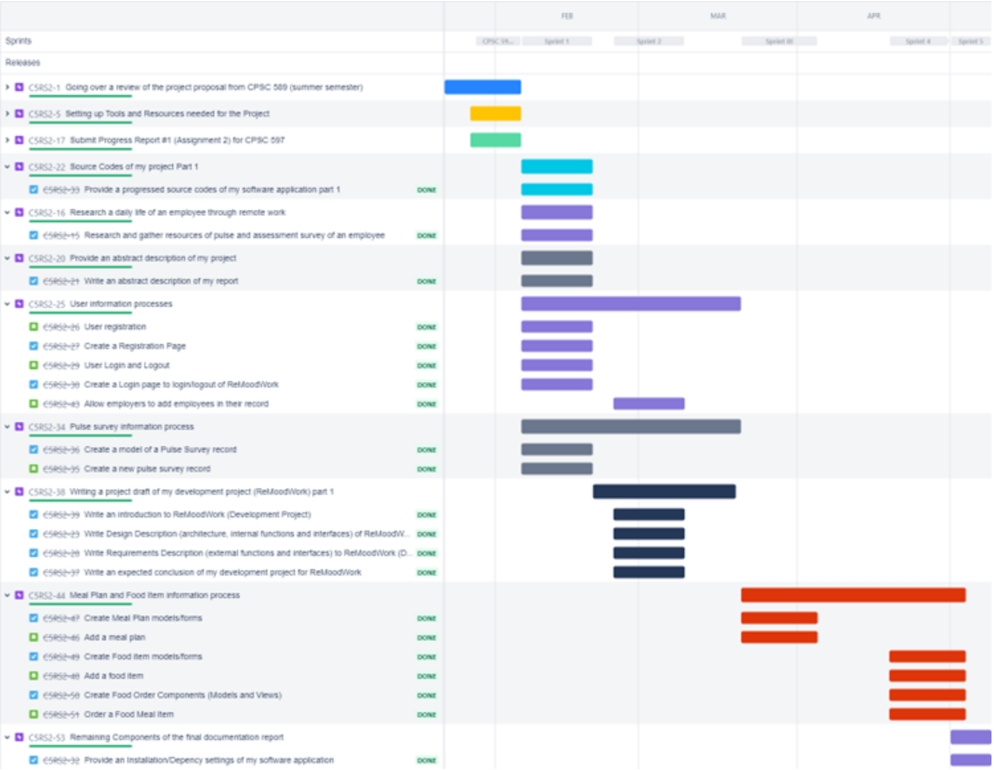
The integration phases of this entire software development have been arranged and identified as a website application used in this project.

Table 2 provides the current project schedule of the ReMoodWork development project shown below:

Table 2 Project Schedule of ReMoodWork

2023	February				March				April				May					Summary	
Tasks:	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	5	Hours	Percent
Planning and Requirements	24	22	23	10	15	20	8	6	10	5	9	4	9					165	28.8%
Design	24	20	19	10	12	13	8	3	10	5	9	4	9					146	25.48%
Coding and Integration	24	23	23	15	21	22												128	22.34%
Integration (continue) and Testing							10	6	14	14	14	12	6					76	13.26%
Write report for running/operating software application													12	8				20	3.49%
Write Final Report														12	13	7		32	5.58%
Demonstrate																	6	6	1.05%
Hours	**72**	**65**	**65**	**35**	**48**	**55**	**26**	**15**	**34**	**24**	**32**	**20**	**36**	**20**	**13**	**7**	**6**	**573**	100.0%

There were some overlaps in completing any unfinished remaining tasks and features discussed from the previous to the next sprint or iteration of this developed project. However, the Jira Software was able to fulfill and adapt Scrum as an Agile Framework tool for handling all user stories, tasks, or requirements used for the next iteration or sprint release planning with the help of a Sprint Backlog used in this project. In addition to Sprint Backlog, it fulfilled user stories, tasks, or requirements with a time-boxed iteration or sprint release plan every two to three weeks to determine the majority of time spent on this development project. **Table 3** provides a release plan used in this project shown below:



Sprints

Releases

- CSRS2-1 Going over a review of the project proposal from CPSC 589 (summer semester)
- CSRS2-5 Setting up Tools and Resources needed for the Project
- CSRS2-17 Submit Progress Report #1 (Assignment 2) for CPSC 597
- CSRS2-22 Source Codes of my project Part 1
  - ~~CSRS2-33~~ Provide a progressed source codes of my software application part 1 DONE
- CSRS2-16 Research a daily life of an employee through remote work
  - ~~CSRS2-15~~ Research and gather resources of pulse and assessment survey of an employee DONE
- CSRS2-20 Provide an abstract description of my project
  - ~~CSRS2-21~~ Write an abstract description of my report DONE
- CSRS2-25 User information processes
  - ~~CSRS2-26~~ User registration DONE
  - ~~CSRS2-27~~ Create a Registration Page DONE
  - ~~CSRS2-29~~ User Login and Logout DONE
  - ~~CSRS2-30~~ Create a Login page to login/logout of ReMoodWork DONE
  - ~~CSRS2-43~~ Allow employers to add employees in their record DONE
- CSRS2-34 Pulse survey information process
  - ~~CSRS2-36~~ Create a model of a Pulse Survey record DONE
  - ~~CSRS2-35~~ Create a new pulse survey record DONE
- CSRS2-38 Writing a project draft of my development project (ReMoodWork) part 1
  - ~~CSRS2-39~~ Write an introduction to ReMoodWork (Development Project) DONE
  - ~~CSRS2-23~~ Write Design Description (architecture, internal functions and interfaces) of ReMoodW... DONE
  - ~~CSRS2-28~~ Write Requirements Description (external functions and interfaces) to ReMoodWork (D... DONE
  - ~~CSRS2-37~~ Write an expected conclusion of my development project for ReMoodWork DONE
- CSRS2-44 Meal Plan and Food Item information process
  - ~~CSRS2-47~~ Create Meal Plan models/forms DONE
  - ~~CSRS2-46~~ Add a meal plan DONE
  - ~~CSRS2-49~~ Create Food Item models/forms DONE
  - ~~CSRS2-48~~ Add a food item DONE
  - ~~CSRS2-50~~ Create Food Order Components (Models and Views) DONE
  - ~~CSRS2-51~~ Order a Food Meal Item DONE
- CSRS2-53 Remaining Components of the final documentation report
  - ~~CSRS2-32~~ Provide an Installation/Depency settings of my software application DONE

Table 3 Release Planning/Roadmap of ReMoodWork

The expected release planning roadmap and a project schedule used in this software development process deliver five total iterations or sprints planned out in this project.

Each iteration or sprint contains user stories that pertain to the functional and nonfunctional requirements described from the requirements description depicted in this report (Please see number three (Requirements Descriptions) used in this report). Furthermore, this project fulfilled 165 hours spent on the functional gathering requirements used in this report. This project also provides story point estimations on determining the complexities and risks for each user story and their functional requirement tasks to satisfy the expected hours spent on the requirement gathering and planning of the project development report. Furthermore, the story point estimations used in this project use a Fibonacci scale (1, 2, 3, 5, 8, 13, 21, …) to measure the range between the simplicity (1) and complex (21 and more) used in this project.

This project uses a Velocity chart report [8] to measure the amount of work spent on each user story and task as story points needed to be committed and finished to achieve its expected projection shown in **Table 4** provided in this project.

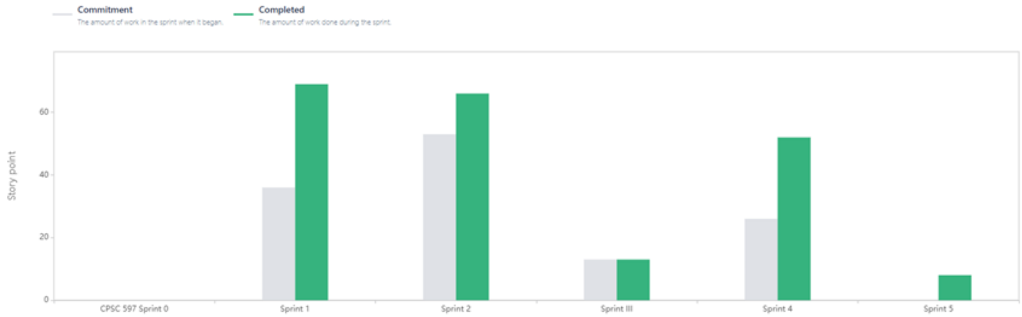


Table 4 Velocity Chart Layout of ReMoodWork

This project provides a descriptive plan for how each iteration or sprint phase was delivered and reviewed with the story estimation points shown below:

5.1.1 Sprint I – Laying the essential groundwork with user registration, authentication, and pulse survey record process

02/06/2023-02/19/2023

This phase covers user stories and tasks to cover the foundational basis of the groundwork used to start developing the software application product of ReMoodWork, beginning with the user registration/authentication and pulse survey records provided in this phase with a possible story point estimations for each user story or task shown below:

Task 1: Research and gather resources of pulse and assessment survey of an employee (No story point estimation):

Researched and gathered necessary information by providing essential components or fields of pulse surveys (work tasks or activities) and assessment surveys (meal plans) used by the employee's informational report.

User Story 1: User registration (Story point - 5):

As a user, I want to register my account, so that I can input my credential information and set my job classification status types between an Employee and an Employer before logging into the ReMoodWork site.

Task 2: Create Registration Page for Users (Story Point - 5):

This task relates to the first user story (User registration) discussed in this phase for creating a registration page for users to input their credential information and job classification type status between Employee and Employer. When users successfully create their account, they will safely redirect back to the home page of ReMoodWork.

User Story 2: User Login and Logout (Story Point - 8):

As a user, I want to log into the home page of ReMoodWork, so that I can visit my list of pulse/assessment survey pages and safely log out of its system.

Task 3: Create a Login page to login/logout of ReMoodWork (Story Point - 8):

This task relates to the second user story (User login and logout) discussed in this phase for creating a login page for users to fulfill their credential information and redirect them to the home page of ReMoodWork containing their list of survey records. This feature also allows users to log out of the system and redirect them to the home page without providing any survey records delivered to the users.

User Story 3: Create a new pulse survey record (Story Point - 21):

As an employee (user), I want to create a new pulse survey record, so that I can keep track with my daily lifestyle activity of my remote work environment and notify this information to my employer everyday.

Task 4: Create a pulse survey record page (Story Point - 21)

This task relates to the third user story (Create a new pulse survey record) discussed in this phase for employees creating their pulse survey record. It is executed by saving the form and storing its instance to the data models of the database table to ensure the response was successfully sent and displayed from the user's request used in the ReMoodWork system application.

5.1.2 Sprint II – Setting up documents for the project draft report of a ReMoodWork development project

02/24/2023-03/09/2023

This phase deliberately focused on setting up the sections of the document to write a project draft report of ReMoodWork's development project. These include writing an introduction report, providing requirements and design descriptions, and adding an expected conclusion report of the overall quality of ReMoodWork used in this report.

There was only one user story and task created in this phase, shown below:

User Story 4: Allow employers to add employees in their record (Story Point – 13):

As an employer, I want to add an employee, so that I can connect with certain employees associated with my current company name.

Task 5: Add employee in an employers' record (Story Point – 13):

This task relates to the fourth user story (Add employee in an employers' record) discussed in this phase for adding employees from the same company name to save records of an employee and let employers have access to their survey records used for keeping up-to-date with them in the future.

5.1.3 Sprint III – Implementing Meal Plan Assessement

03/21/2023-04/16/2023

This phase provides estimated story points for a user story and task to set up and create meal plan assessments made by an employee. However, this phase did not have a user story or task plan to construct a food item that deals with restaurants or food recipes but provided its setup plan for Sprint IV as the next sprint iteration.

User Story 5: Add meal planning assessment (Story Point – 8):

As an employee, I want to add a meal plan, so that I can input my health and dietry contents before selecting food items based on my personal information of my data.

Task 6: Create Meal Plan setup (Story Point – 8):

This task relates to the fifth user story (Add meal planning assessments) discussed in this phase for employees adding their meal plan assessments using model, form, and view to test and verify that each component worked through the website of ReMoodWork.

5.1.4 Sprint IV – Implementing Food Items and Order for employees ordering food meal items

04/19/2023 – 5/1/2023

This phase provides estimated story points for a user story and task to set up and create food meal items made by employers and employees ordering food meal items based on their meal planning assessment information. This phase also delivers a user story and task plan to produce and order food meal items from selecting it as a restaurant or recipe as their choices used in ReMoodWork.

User Story 6: Add a food item (Story Point – 13):

As an employer, I want to add a food item to include meals made for the employee to select from a restaurant name or food recipe of their choice.

Task 7: Create Food item (Story Point – 13):

This task relates to the sixth user story (Add a food item) discussed in this phase for employers adding a meal item using model, form, and view to test and verify that each component worked on the ReMoodWork website.

User Story 7: Order a food meal item (Story Point – 13):

As an employee, I want to order a food item from the website, so that I can place my meal from a restaurant or recipe based on my meal planning assessment profile.

Task 8: Create Food Order Components (Story Point – 13):

This task relates to the seventh user story (Order a food meal item) discussed in this phase for employees to select and place an ordered meal, to only use model and view to test and verify that each component worked on the ReMoodWork website.

5.1.5 Sprint V – Finishing remaining parts for the final documentation report of ReMoodWork

05/01/2023 – 05/08/2023

This phase focuses on finishing the remaining parts needed to wrap up the final documentation report of ReMoodWork, including writing implementations, installation instructions, operating instructions, and achieved features/functionalities of the project, as well as providing recommended enhancements made for the final report.

5.2 Testing

This section covers the testing phase of ReMoodWork's software application platform during its development process and researched analysis on how each part of the product shall be tested and verified for its featured functionalities pertained by user stories and requirements used in the project.

This project's repository provides test suites and files stored in a test folder for each application name used in the project. This section contains test cases and evaluated results using a grey-box testing strategy that provides black-box (unit testing) and a few white-box (acceptance testing) strategies used to verify a software application product of ReMoodWork.

This project determines essential functionalities and features for each test scenario infrastructure of all application names used in this project. Each application name tests models, forms, and views discussed in the design requirements in this project report (Please see the Design Requirements section that covers the Model-View-Template architecture pattern of ReMoodWork). Furthermore, the models and forms of each application name provide an evaluation using a unit testing technique related to the database portion of ReMoodWork's software development system. Compared to testing models and forms used in this project, the views of each application name provide an evaluation using an acceptance testing technique used to handle HTTP requests made by a client to provide interactions and experiences happening in ReMoodWork's software system. The following section covers test cases and results for test scenarios covering each application product name used in the project.

5.3 Test Cases and Results for each Named Application of ReMoodWork's software application

5.3.1 Testing Users' Application

Unit Testing

Test Scenario 1: Create User's Model and Registration Form

Required Fields: Username, Password, Full Name, Email Address, Company Name, Job Classification (Employee or Employer Choice)

This test scenario covers unit testing involved in creating data models and saving data registration forms of a user. Furthermore, the test file names test_users_models.py (data models) and test_user_registration_form.py (registration forms) shall require users to enter their six required attribute fields for registering their information through a software application system of the ReMoodWork environment. The six attribute fields required for user data models and registration forms (seven attribute fields due to password confirmation for registration forms) are username, password, full name, email address, company name, and job classification choice (Employee or Employer) used in this test scenario of a user's application name for the ReMoodWork product. These test cases pertain to the six required fields to verify that the user's data models and registration forms can be satisfied enough for its functionalities and expected output results for ReMoodWork's software application product. The user's data models make it simple enough to use valid inputs and verify that the model gets created in a dummy database table of ReMoodWork. However, it is challenging enough to determine different input values and outcomes from the user's registration form where they will be saved and submit a response back to the user. Therefore, by preventing any test failures from happening with the user registration form of this project, it shall cover specific valid and invalid inputs to test its saving data creation and validation of test_user_registration_form.py used in this test scenario form.

Test Summary of test_users_models.py (User Data Model)

Test Case ID:	Test Name:	Test Fields:	Input Values	Input Type	Expected Behavior	Test Observation	Test Result
1	Create a User Data Model	username, password, full name, email address, company name, job classification choice (Employer or Employee)	johnsmith, test123!, John Smith, johntsmith@gmail.com, Google, EMPLOYEE	Valid input	Successfully created a user data model	Successfully created a user data model	Pass
2	Create an Employee Data Model	username, password, full name, email address, company name, job classification choice (Employer or Employee)	johnsmith, test123!, John Smith, johntsmith@gmail.com, Google, EMPLOYEE	Valid input	Successfully created an employee data model **due to the value 'EMPLOYEE' set as job classified choice**	Successfully created an employee data model **due to the value 'EMPLOYEE' set as job classified choice**	Pass
3	Create an Employer Data Model	username, password, full name, email address, company name, job classification choice (Employer or Employee)	edwardsilverman, testing123!, Edward Silverman, esilverman@gmail.com, Google, EMPLOYER	Valid input	Successfully created an employer data model **due to the value 'EMPLOYER' set as job classified choice**	Successfully created an employer data model **due to the value 'EMPLOYER' set as job classified choice**	Pass

Test summary of test_user_registration_form.py (User registration)

Test Case ID:	Test Name:	Test Fields:	Input Values	Input Type	Expected Behavior	Test Observation	Test Result
1	Create User Registration Form as an Employee	username, password1, password2, full name,	johnsmith, test123!, test123!, John Smith, johntsmith@gmail.com, Google, EMPLOYEE	Valid input	Successfully saved the user's data from the	Successfully saved the user's data from the	Pass

		email address, company name, job classification choice (Employer or Employee)			registered form.	registered form.	
2	Create User Registeration Form as an Employer	username, password1, password2, full name, email address, company name, job classification choice (Employer or Employee)	jamesjohnson, jamestest456!, jamestest456!, James S. Johnson, jamesjohnson@gmail.com, Google, EMPLOYER	Valid input	Successfully saved the user's data from the registered form as an Employer	Successfully saved the user's data from the registered form as an Employer	Pass
3	Unable to save user registration information	username, password,	billysilver, testname15	Invalid input	Unsuccessfully saved user's data from the registered form **due to the missing field values not entered with full name, email address, company name, and job classification choice (Employee or Employee)**	Unsuccessfully saved user's data from the registered form **due to the missing field values not entered with full name, email address, company name, and job classification choice (Employee or Employee)**	Pass
4	Username already exists	username, password1, password2, full name, email address, company name, job classification	johnsmith, test123!, test123!, Billy Silverstone, johntsmith@gmail.com, Google, EMPLOYEE	Invalid input	Unsuccessfully saved user's data from the registered form **due to the same username value as test case ID 1**	Unsuccessfully saved user's data from the registered form **due to the same username value as test case ID 1**	Pass

		choice (Employer or Employee)					
5	Full name already exists	username, password1, password2, full name, email address, company name, job classification choice (Employer or Employee)	billysilver, testname15, testname15, John Smith, johntsmith@gmail.com, Google, EMPLOYEE	Invalid input	Unsuccessfully saved user's data from the registered form **due to the same full name value as test case ID 1**	Unsuccessfully saved user's data from the registered form **due to the same full name value as test case ID 1**	Pass

Test Scenario 2: User Login Form

<u>Required fields: username and password</u>

This test scenario covers unit testing involved in authenticating users' credentials from the Login Form of ReMoodWork. Furthermore, the test file name test_login_user_form.py (login form) shall require users (regardless of their job classification status between employees and employers) to input their username and password as two values used for logging into the website page of ReMoodWork. Users must first create their account before becoming authenticated and logging into the main homepage website of ReMoodWork. This test scenario covers three valid inputs used for evaluating and saving the submission login form from the user.

Test Summary of test_login_user_form.py (User Login)

Test Case ID:	**Test Name:**	**Test Fields:**	**Input Values**	**Input Type**	**Expected Behavior**	**Test Observation**	**Test Result**
1	User Created	username, password, full name	jamiesmith, testing123, Jamie Smith	Valid input	Successfully created user account	Successfully created user account	Pass
2	User Login with correct username and password	username, password	jamiesmith, testing123	Valid input	Successfully authenticated a user with correct username and password	Successfully authenticated a user with correct username and password	Pass
3	User Login with incorrect username	username, password	timtest, timisthebest123	Valid input	Unsuccessfully authenticated a user due to the wrong username and	Unsuccessfully authenticated a user with incorrect username and	Pass

	and password				password provided by the user	password	

Acceptance Testing

Test Scenario 3: Testing User's View of ReMoodWork

<u>Required fields: username, password (2x), full name, email address, company name, job classification (Employee or Employer choice)</u>

This acceptance testing covers the view functionality of ReMoodWork's software application system based on the HTTP responses sent from the user's request. Furthermore, this test scenario relates to testing cases covered by the user's data model and input forms addressed by unit testing. These views surrounding acceptance testing based on handling HTTP requests include user registration, user login/logout, and user registration as an employer. It also covers acceptance testing for employers to add and check the employee's records based on the same company name.

Test summary of test_users_view.py (Users View of ReMoodWork)

Test Case ID:	**Test Name:**	**Test Fields:**	**Input Values**	**Input Type**	**Expected Behavior**	**Test Observation**	**Test Result**
1	User Registration View	username, password1, password2, full name, email address, company name, job classification choice (Employer or Employee)	johnsmith, test123!, test123!, John Smith, johntsmith@gmail.com, Google, EMPLOYEE	Valid Input	Should be able to successfully register users straight out of the registration page of ReMoodWork with the 302 as a status code redirect to home page of ReMoodWork.	Successfully register users straight out of the registration page of ReMoodWork with the 302 as a status code redirect to home page of ReMoodWork.	Pass
2	User Login and Logout View	username, password	johnsmith, test123!	Valid Input	Should be successfully able to login and logout of the ReMoodWork website made with status code 302 to redirect to the home page after logging in	Successfully able to login and logout of the ReMoodWork website made with status code 302 to redirect to the home page after logging in and 200 for	Pass

					and 200 for logging out of the system.	logging out of the system.	
3	User Registration View as an Employer	username, password1, password2, full name, email address, company name, job classification choice (Employer or Employee)	jamesjohnson, jamestest456!, jamestest456!, James S. Johnson, jamesjohnson@gmail.com, Google, EMPLOYER	Valid Input	Should be able to successfully register users as employers straight out of the registration page of ReMoodWork with the 302 as a status code redirect to the home page of ReMoodWork.	Successfully register users as employers straight out of the registration page of ReMoodWork with the 302 as a status code redirect to the home page of ReMoodWork.	Pass
4	Add an Employee	username, password1, password2, full name, email address, company name, job classification choice (Employer or Employee)	jamesjohnson, jamestest456!, jamestest456!, James S. Johnson, jamesjohnson@gmail.com, Google, EMPLOYER (For Employer) johnsmith, test123!, test123!, John Smith, johntsmith@gmail.com, Google, EMPLOYEE (For Employee)	Valid Input	Should be able to successfully allow an employer to save and add an employee of the same company name.	Successfully allow an employer to save and add an employee of the same company name.	Pass

5.3.2 Work Record's Application

Unit Testing

Test Scenario 1: Create Pulse Survey Models and Forms

<u>Required fields: activity name, activity type, number of hours spent on the activity, emotional rating scale, work stressor status, activity created, employee, and activity description (optional)</u>

This test scenario covers unit testing involved in creating and saving pulse survey records made by a registered employee. Furthermore, the test file names test_pulse_survey_models.py (for pulse survey models) and test_pulse_survey_creation_form.py (for creating pulse survey forms) shall allow employees to enter five required fields (one optional field) for completing their pulse surveys used to present in ReMoodWork's software system. The five required attributes required for employees creating pulse surveys using both models and forms are activity name, activity type, number of hours spent on the activity, emotional rating scale, and work stressor status, with only activity description being the optional field used in this test scenario as part of workrecord's application name used in ReMoodWork. Furthermore, test cases related to the five data attributes presented for testing pulse survey models and forms can satisfy enough for its functionalities and expected output results for ReMoodWork's software application product. Similar to strategizing in creating data models and registration forms for a user (Please see test scenarios of the user's application), the same applies to creating pulse surveys based on its data model and forms made for a registered employee's usage of ReMoodWork software. The data models of pulse surveys deliver simplicities in handling valid inputs and verifying that the model gets created in a dummy database table of ReMoodWork. However, there are some circumstances to look at defining different input values made for saving and evaluating data validation in submitting pulse survey records forms made by an employee. To prevent any test failures to create pulse survey records by employees, it covers specific valid and invalid inputs to test its saving data creation and validation of test_pulse_survey_creation_form.py used for testing the test results of the pulse survey forms.

Test summary of test_pulse_survey_models.py (Creating Pulse Survey data models)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employee (required for creating Pulse Surveys)	(Must use User data model first) username, password, full name, email, company name, job classification choice	mikerandy, anothertest555!, Mike Randy, mikerandy@test.com, Amazon, EMPLOYEE	Valid Input	The system should successfully create an employee from a created user used in ReMoodWork's database model.	Successfully created an employee from a created user used in ReMoodWork's database model.	Pass
2	Create Pulse Survey Model	activity name, activity type, number of hours, emotional rate status, activity description (optional), work stressor status, activity created, employee	***(Brackets represents a map out selection choices made by an employee)*** Implemented campaign feature test cases for ecommerce website, WT ***[Work task]***, 10, taking_break_or_vacation ***[\N{smiling face with sunglasses}]***, I have created 5 additional test cases of a campaign feature used for ecommerce website (optional), NO ***[No]***, today's date (like	Valid input	The system should successfully create a new pulse survey record as a data model.	Successfully created a new pulse survey record as a data model.	Pass

			3/11/2023), same employee from previous test case (Mike Randy)				
3	Create another Pulse Survey Model	activity name, activity type, number of hours, emotional rate status, activity description (optional), work stressor status, activity created, employee	***(Brackets represents a map out selection choices made by an employee)*** Taco Party, TB ***[Take a break]***, 1, taking_break_or_vacation ***[\N{smiling face with sunglasses}]***, Celebrating taco party with a few of my coworkers at a break room! (optional), NO ***[No]***, today's date (like 3/11/2023), same employee from previous test case (Mike Randy)	Valid Input	The system should successfully create another new pulse survey record made by the same employee used from the two previous test cases for this test scenario.	Successfully created another new pulse survey record made by the same employee from the two previous test cases for this test scenario.	Pass

Test summary of test_pulse_survey_creation_form.py (Pulse Survey Creation Form)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employee (required for creating Pulse Surveys)	(Must use User data model first) username, password, full name, email, company name, job classification choice	mikerandy, anothertest555!, Mike Randy, mikerandy@test.com, Amazon, EMPLOYEE	Valid input	The system should successfully create an employee from a created user used in ReMoodWork's database model.	Successfully created an employee from a created user used in ReMoodWork's database model.	Pass
2	Create Pulse Survey Records Form	activity name, activity type, number of hours, emotional rate status, activity description (optional), work stressor status, activity	***(Brackets represents a map out selection choices made by an employee)*** Created specifications report for campaign pages for an ecommerce, WT ***[Work task]***, 10, sleeping ***[\N{sleeping face}]***, Wrote two descriptive features	Valid Input	The form should successfully save and create a new pulse survey record from its form.	Successfully save and create a new pulse survey record from its form.	Pass

		created, employee	deliveres for campaign pages of an ecommerce (optional), YES ***[Yes]***, today's date (like 3/11/2023), same employee from previous test case (Adam Smart)				
3	Activity name already created	activity name, activity type, number of hours, emotional rate status, activity description (optional), work stressor status, activity created, employee	Same input values from test case 2 used in this test scenario	Invalid Input	The form should unsuccessfully save and create a new pulse survey forms **due to the same activity name already created.**	Unsuccessfully save and create a new pulse survey forms **due to the same activity name already created.**	Pass
4	Required fields to create pulse survey forms	activity name, number of hours	Office party, 3	Invalid Input	The form should unsuccessfully save and create a new pulse survey form due to the other missing fields needed to create a pulse survey.	Unsuccessfully save and create a new survey forms due to the other missing fields needed to create a pulse survey.	Pass

Test Scenario 2: Create Meal Plan Assessment Models and Forms

<u>Required fields: calories, dietary restrictions, goal, allergy, budget, cuisine, and employee</u>

This unit testing covers creating and saving meal plan assessments by a registered employee. Furthermore, the test file names test_meal_plan_model.py (for meal plan assessment models) and test_meal_plan_creation_form.py (for creating meal plan assessment forms) shall allow employees to enter six required fields for completing their meal plan assessments used to present in ReMoodWork's software system. The six attributes required for employees to create their meal plan assessment using both models and forms are calories, dietary restrictions, goal, allergy, budget, and cuisine used in this test scenario as part of a workrecord's application of ReMoodWork. Furthermore, test cases related to the six data attributes presented to test and verify meal plan assessment models and forms can satisfy the functionalities and expected output results of ReMoodWork's software application product. Meal planning assessments must be created only once (one-to-one relationship) by an employee in ReMoodWork. Similar to other data models from previous test scenarios, this test scenario also focuses on defining different valid and invalid input values to save data creation and validation of test_meal_plan_creation_form.py used for testing test results of meal planning assessment records.

Test summary of test_meal_plan_model.py (Meal Plan Assessment Models)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employee (required for creating Meal Plan Assessment)	(Must use User data model first) username, password, full name, email, company name, job classification choice	mikerandy, anothertest555!, Mike Randy, mikerandy@test.com, Amazon, EMPLOYEE	Valid Input	The system should successfully create an employee from a created user used in ReMoodWork's database model.	Successfully created an employee from a created user used in ReMoodWork's database model.	Pass
2	Create a Meal Planning Model	calories, dietary restrictions, goal, allergy, budget, cuisine	190, gluten-free, To lose weight by 30 lbs, wheat, 20.00, American	Valid input	The system should successfully create a new meal planning assessment as a data model.	Successfully created a new meal plan assessment record as a data model.	Pass
3	Create another Meal Planning model	calories, dietary restrictions, goal, allergy, budget, cuisine	220, vegetarian, To lose weight by 10 lbs, meat, 30.00, Greek	Invalid Input	The system should not successfully create another meal planning assessment as a data model, due to a one-to-one relationship.	Successfully not creating another meal planning assessment as a data model, due to a one-to-one relationship.	Pass

Test summary of test_meal_plan_creation_form.py (Meal Plan Assessment Creation Form)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employee (required for creating	(Must use User data model first) username, password,	adamsmart, anothertest666!, Adam Smart, adamsmart@test.com,	Valid Input	The system should successfully create an	Successfully created an employee from a created user	Pass

	Meal Plan Assessment)	full name, email, company name, job classification choice	Apple, EMPLOYEE		employee from a created user used in ReMoodWork's database model.	used in ReMoodWork's database model.	
2	Create a Meal Planning Form	calories, dietary_restrictions, goal, allergy, budget, cuisine	300, gluten-free, To lose weight by 30 lbs, wheat, 20.00, American	Valid input	The form should save, and create a new meal planning assessment form.	Successfully saves and creates a new meal planning assessment form.	Pass
3	No negative numbers of the calorie information for meal planning assessment form	calories, dietary restrictions, goal, allergy, budget, cuisine	-1, gluten-free, To lose weight by 30 lbs, wheat, 1.00, American	Invalid input	The form should not save and create a new meal plan assessment record from its form due to a negative number as a value used in calories.	Successfully not saved and create a new meal plan assessment record from its form due to a negative number as a value used in calories.	Pass
4	Required fields for creating meal planning assessment forms	allergy, cuisine	peanuts, Asian	Invalid Input	The form should unsuccessfully save, and create a new meal plan assessment form due to the other missing fields.	Unsuccessfully save and create new meal plan assessment forms due to the other missing fields.	Pass

Test Scenario 3: Create Food Item Models and Forms

<u>Required fields: Food name, description (optional), price (optional if recipe), cuisine type, food item type, recipe_url (optional if restaurant), the restaurant name (optional if recipe), calories, dietary restrictions, allergy, an image of a food meal item (optional), and employer</u>

This unit testing covers creating and saving food items by a registered employer. The test file test_food_item_model.py covers the food item model, and test_food_item_creation_form.py covers food item creation forms. These test script files evaluate whether an employer can enter six required and five optional fields to create a new food item as a restaurant or recipe meal. These fields contain the food name, description, price, cuisine type, food item type, recipe URL, restaurant name, calories, dietary restrictions, allergy, and an image of a food meal item needed to verify and cover test cases of food items between models and forms. Therefore, it shall satisfy food item features and expected output results of ReMoodWork. This test scenario also defines different valid and invalid input values to save data creation and validation of test_food_item_creation_form.py used for testing test results of food meal items.

Test summary of test_food_item_model.py (Food Item Model)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employer (required for creating Food Item)	(Must use User data model first) username, password, full name, email, company name, job classification choice	tobeysmith, testingemployer123!, Tobey Smith, tobeysmith@test.com, Amazon, EMPLOYER	Valid Input	The system should successfully create an employer from a created user used in ReMoodWork's database model.	Successfully created an employer from a created user used in ReMoodWork's database model.	Pass
2	Create a Food Item Model with Restaurant	food_name, description, price, cuisine_type, food_item_type, restaurant_name, calories, dietary_restrictions, allergy, food_meal_image	Ribeye Steak, This steak is cooked with medium-rare and provides cilantro spices and buttery flavor., 50.00, american, restaurant, Jim Steakhouse Restaurant, 300, gluten-free, beef, Example_Steak.jpg	Valid input	The system should successfully create a new food item with the restaurant as a data model.	Successfully created a new food item with the restaurant as a data model.	Pass
3	Create a Food Item Model with Recipe	food_name, description, cuisine_type, food_item_type, recipe_url, calories, dietary_restrictions, allergy	Carne Asada, This recipe will be good if you are a meat lover, mexican, recipe, https://loseweightbyeating.com/carne-asada-recipe/, 250, gluten-free, beef	Valid Input	The system should successfully create a new food item with the recipe as a data model.	Successfully created a new food item with the recipe as a data model.	Pass

Test summary of test_food_item_creation_form.py (Food Item Creation Form)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employer (required for creating Meal Plan Assessment)	(Must use User data model first) username, password, full name, email,	tobeysmith, testingemployer123!, Tobey Smith, tobeysmith@test.com, Amazon, EMPLOYER	Valid Input	The system should successfully create an employer from a created user	Successfully created an employer from a created user used in	Pass

		company name, job classification choice			used in ReMoodWork's database model.	ReMoodWork's database model.	
2	Create a Food Item Form with Restaurant	food_name, description, price, cuisine_type, food_item_type, restaurant_name, calories, dietary_restrictions, allergy, food_meal_image	Smoked salmon, Cooked and marinated with lemon juice, 20.00, american, restaurant, Dr.Dock\'s Seafood Restaurant, 200, gluten-free, shellfish, Example_Salmon.jpg	Valid input	The form should save and create a new food item form with a restaurant type.	Successfully saved and create a new food item form with a restaurant type from its form.	Pass
3	Create a Food Item Form with Recipe	food_name, description, cuisine_type, food_item_type, recipe_url, calories, dietary_restrictions, allergy	Carne Asada, This recipe will be good if you are a meat lover, mexican, recipe, https://loseweightbyeating.com/carne-asada-recipe/, 250, gluten-free, beef	Valid Input	The form should save and create a new food item form with a recipe type.	Successfully saved and create a new food item form with a recipe type from its form.	Pass
4	Required fields for creating food item forms	food_name, description	Goulash, You will enjoy this meal if you like meat and vegetables	Invaid Input	The form should unsuccessfully save and create a new food item form due to other missing fields.	Unsuccessfully save and create a new food item form due to other missing fields.	Pass
5	Invalid choices for creating food item forms (cuisine_type, food_item_type, dietary_restictions, allergy)	food_name, description, cuisine_type, food_item_type, calories, dietary_restictions, allergy	Strawberry Crepes, Can be served during breakfast, lunch, or dinner, french, other, 150, glutenfree, strawberry	Invalid Input	The form should unsuccessfully save and create a new food item due to invalid values of choices found in each one (cuisine_type, food_item_type, dietary_restictions, allergy)	Unsuccessfully save and create a new food item form due to invalid values of choices found in each one (cuisine_type , food_item_type, dietary_restictions, allergy)	Pass
6	Calories negative	food_name, description, price,	Smoked salmon, Cooked and marinated with lemon juice,	Valid Input	The form should	Unsuccessfully save and	Pass

	number	cuisine_type, food_item_type, restaurant_name, calories, dietary_restrictions, allergy	20.00, american, restaurant, Dr.Dock\'s Seafood Restaurant, -1, gluten-free, shellfish		unsuccessfully save and create a new food item due to not accepting negative number of the calories attribute	create a new food item due to not accepting negative number of the calories attribute	

Test Scenario 4: Create Order Models to place a Food Meal Item

Required fields: employee (Must have Employee data model), meal plan (Must have Meal Plan data model), and food items (Must have Employer and FoodItem data model)

This unit testing covers creating an Order data model for placing a food meal order by an employee through the test file name test_order_food_item_model.py. This test script requires an Employee data model to register a one-to-many relationship to produce one or more food meal orders. It also requires an Employer and FoodItem data model for registering and creating multiple food items for employees to choose from their meal planning assessment page in ReMoodWork. This test scenario also does not need to require and verify a form for employees ordering food meal items due to saving and producing for each Order data model.

Test summary of test_order_food_item_model.py (Order a Food Item)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	Register Employer (required for creating Food Item)	(Must use User data model first) username, password, full name, email, company name, job classification choice	tobeysmith, testingemployer123!, Tobey Smith, tobeysmith@test.com, Amazon, EMPLOYER	Valid Input	The system should successfully create an employer from a created user used in ReMoodWork's database model.	Successfully created an employer from a created user used in ReMoodWork's database model.	Pass
2	Register Employee (required for creating Order data model)	(Must use User data model first) username, password, full name, email, company name, job classification choice	mikerandy, anothertest555!, Mike Randy, mikerandy@test.com, Amazon, EMPLOYEE	Valid input	The system should successfully create an employee from a created user used in ReMoodWork’s database model.	Successfully created an employee from a created user used in ReMoodWork’s database model.	Pass
3	Register	calories,	400, “Gluten-	Valid Input	The system should	Successfully	Pass

	Meal Plan (required for achieving an algorithm in filtering Food Item data models) [Must acquire Employee model]	dietary_restrictions, goal, allergy, budget, cuisine	free, vegetarian”, To lose weight by 30 lbs, “Wheat, peanuts, shellfish”, 20.00, “American, mexican”		successfully create a meal planning assessment used in ReMoodWork’s database model.	created a meal planning assessment used in ReMoodWork’s database model.	
4	Register Two Food Items (required for Employees creating an Order Meal)	food_name, description, price, cuisine_type, food_item_type, recipe_url, restaurant_name, calories, dietary_restrictions, allergy (for both input values of Food Item data model), food_meal_image (for the first Food Item with ‘Smoked Salmon’)	Smoked Salmon, Cooked and marinated with lemon juice, 20.00, american, restaurant, Dr.Dock\'s Seafood Restaurant, 200, gluten-free, soy, Example_Salmon.jpg (for Restaurant Food Item data model) Texas-Style Beef Brisket, This recipe is the best if you love BBQ, american, recipe, https://www.tasteofhome.com/recipes/texas-style-beef-brisket/, 381, gluten-free, beef (for Recipe	Valid Input	The system should successfully create two food meal items used in ReMoodWork’s database model.	Successfully created two food meal items used in ReMoodWork’s database model.	Pass

			Food Item data model)				
5	Filter and Order Food Item	employee, meal_plan, food_items	Employer, Employee, Meal Plan, and two Food Items data model (Please see "Register Employer", "Register Employee", "Register Meal Plan", and "Register Two Food Items"). The Food Item selected is 'Smoked Salmon'	Valid Input	The system should successfully filter food items and order a selected food item from an employee.	Successfully filter food items and order a selected food item from an employee.	Pass

Acceptance Testing

Test Scenario 5: Testing Pulse Survey View of ReMoodWork

Required fields: activity name, activity type, number of hours spent on the activity, emotional rating scale, work stressor status, activity created, and activity description (optional)

This acceptance testing covers the view functionality of ReMoodWork's software application system for the HTTP responses sent from employees' requests on creating their pulse survey records. Furthermore, this test scenario relates to software testing cases covered by the pulse survey data model and input forms addressed by unit testing. These views surround acceptance testing based on handling HTTP requests, including an overview and creating pulse surveys on the page made by an employee.

Test summary of test_pulse_survey_creation_view.py (Pulse Survey Creation View)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	View Pulse Survey Page	***(As an employee)*** username, password (required for logging into ReMoodWork)	adamsmart, anothertest666!	Valid Input	Should be able to successfully display list of pulse survey records presented to	Successfully display list of pulse survey records presented to employees	Pass

					employees.		
2	View Pulse Survey Create Page	***(As an employee)*** username, password, activity name, activity type, number of hours, emotional rate status, activity description (optional), work stressor status, activity created	adamsmart, anothertest666!, Taco Tuesday, TB ***[Take a break]***, 2, taking_break_or_vacation ***[\N{smiling face with sunglasses}]***, Had a blast attending taco party with my coworkers in a breakout room, NO ***[No]***, today's date (like 3/11/2023)	Valid Input	Employees should be able to successfully create their pulse survey as a view response.	Successfully created pulse survey as a view response from an employee	Pass

Test Scenario 6: Testing Meal Plan View of ReMoodWork

Required fields: calories, dietary restrictions, goal, allergy, budget, and cuisine

This acceptance testing covers the view functionality of ReMoodWork's software application system for the HTTP responses sent from employees' requests on creating their meal planning assessment records. Furthermore, this test scenario relates to software testing cases covered by the meal plan assessment's data model and input forms addressed by unit testing. These views surround acceptance testing based on handling HTTP requests, including an overview and creating meal planning assessments on the page by an employee.

Test summary of test_meal_plan_creation_view.py (Meal Plan Creation View)

Test Case ID:	**Test Name:**	**Test Fields:**	**Input Values:**	**Input Type:**	**Expected Behavior:**	**Test Observation:**	**Test Result:**
1	View Meal Plan Page	***(As an employee)*** username, password (required for logging into ReMoodWork)	adamsmart, anothertest666!	Valid Input	Should be able to successfully display a meal planning assessment record presented to employees.	Successfully display a meal planning assessment record presented to employees.	Pass
2	View Meal Plan Create Page	***(As an employee)*** username, password, calories, dietary	adamsmart, anothertest666!, 300, gluten-free, To lose weight by 30 lbs, wheat, 20.00, American	Valid Input	Employees should be able to successfully create their	Successfully created meal planning assessment as a view	Pass

		restrictions, goal, allergy, budget, cuisine			meal plan assessment as a view response.	response from an employee.	

Test Scenario 7: Testing Food Item Creating View in ReMoodWork

Required fields: Food name, description (optional), price (optional if recipe), cuisine type, food item type, recipe URL (optional if restaurant), the restaurant name (optional if recipe), calories, dietary restrictions, allergy, and an image of a food meal item

This acceptance testing covers the view functionality of ReMoodWork's software application system for the HTTP responses sent from employers' requests on creating new food items based on a restaurant or recipe of the meal. Furthermore, this test scenario relates to software testing cases covered by the food item's data model and input forms addressed by unit testing. These views surround acceptance testing based on handling HTTP requests, including an overview and creating food meal items on the page by an employer.

Test summary of test_food_item_creation_view.py (Food Item Creation View)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	View Food Item Create Page	***(As an employer)*** username, password, food_name, description, price, cuisine_type, food_item_type, restaurant_name, calories, dietary_restrictions, allergy, food_meal_image	tobeysmith, testingemployer123!, Smoked salmon, Cooked and marinated with lemon juice, 20.00, american, restaurant, Dr.Dock\'s Seafood Restaurant, 200, gluten-free, shellfish, Example_Salmon.jpg	Valid Input	Employers should be able to create their food item as a view response successfully.	Successfully created food items as a view response from an employer.	Pass

Test Scenario 8: Testing Order View to place a Food Meal Item in ReMoodWork

Required fields: employee (Must have Employee data model), meal plan (Must have Meal Plan data model), and food items (Must have Employer and FoodItem data model)

This acceptance testing covers the view functionality of ReMoodWork's software application system for the HTTP responses sent from employees' requests to order food items based on health and dietary information created from their meal plan assessment. This test scenario relates to test cases covered by utilizing the Order data model to place a food meal item by an employee achieved in unit testing. This view covers acceptance testing based on handling HTTP requests for a page that allows employees to order their food meal items.

Test summary of test_order_food_item_view.py (Order View for placing Food Meal Items)

Test Case ID:	Test Name:	Test Fields:	Input Values:	Input Type:	Expected Behavior:	Test Observation:	Test Result:
1	View Order a Food Meal Item Page	employee, meal_plan, food_items	Employer, Employee, Meal Plan, and two Food Items data models. The Food Item selected for ordering a meal is 'Smoked Salmon.'	Valid Input	Employees should be able to order their food meal items as a view response successfully.	Successfully ordered a food meal item as a view response from an employee.	Pass

6. Installation Instructions

This project provides steps to install other software applications and module libraries shown below:

1. Download the project's zip folder and extract all contents of the project folder to your local directory.
2. Install Python 3.10.6 or higher (https://www.python.org/downloads/)
3. After placing the project folder in your local directory, there are a couple of instructions to install modules or dependencies needed for running the project (Note that my-django-env is used an example to set up a virtual environment in Python).
    a. After installing Python to your OS drive (Windows, Ubuntu, Mac, etc.), go to "ReMoodWork-master" folder using a terminal (highly recommend Ubuntu Linux) or an IDE environment (highly recommend PyCharm) and began creating a Python virtual environment name of your choice using this command in the terminal:
        - `python3 -m venv my-django-env`

    b. Once the virtual environment set up is finished, then activate its current environment of Python by using this command in the terminal:
        - `source my-django-env/bin/activate`

    c. Check for a Python version in the terminal: `python --version`

    d. Use the `deactivate` command if you want to shut off your current virtual environment of Python in the terminal.

    e. Install other modules/libraries (It includes Django and is required to run the ReMoodWork server) of a current Python virtual environment using requirements.txt found in this command in the terminal: `pip install -r requirements.txt`

    f. Lists all of the modules/libraries and their versions used in the current virtual environment of Python in the terminal:
        - `pip freeze`
        - `pip freeze --local`

    g. After installing the necessary modules/libraries found from requirements.txt, reactivate your Python virtual environment: `source my-django-env/bin/activate` and check the current version of Django using this command in the terminal:
        - `python -m django --version`

    h. Once you have Django installed to the project, then follow the rest of the steps to complete installing parts and components of Django shown below:

I. Execute this command that changes from “ReMoodWork-master” directory to “remoodwork” directory in the terminal: `cd remoodwork/`

II. Execute `python manage.py makemigrations` command to install db.sqlite3 (Note that SQLite is a required database needed in this project) in the terminal.

III. Execute `python manage.py migrate` command to install and load database models of Django and ReMoodWork in the terminal.

IV. Run the Django server of ReMoodWork (Project Name) project using this command: `python manage.py runserver` to launch the website in the terminal.

  - You can also run the server with PyCharm IDE through this link: https://www.youtube.com/watch?v=WluSpfSMj2Y

7. Operating Instructions

This section provides Django commands that are helpful to use in this project, shown below:

A. (Optional) Show a list of url routes of ReMoodWork using this command:
  - `python manage.py show_urls | grep 'remoodwork'`

B. (Optional but highly recomended) Create an admin using this command:
  - `python manage.py createsuperuser`
  - Tutorial Reference: https://docs.djangoproject.com/en/1.8/intro/tutorial02/

C. Run test command to evaluate ReMoodWork used through Django (must perform in “remoodwork” directory):
  - Run all tests of ReMoodWork applications (users, workrecords, etc.):
    - `python manage.py test .`
  - Run all tests of ReMoodWork’s users applications:
    - `python manage.py test users`
  - Run all tests of ReMoodWork’s workrecords applications:
    - `python manage.py test workrecords`
  - Run individual test script file of ReMoodWork’s workrecords applications:
    - `python manage.py test workrecords. tests.<test_script>`
  - Run individual test script file of ReMoodWork’s users applications:
    - `python manage.py test users. tests.<test_script>`
  - Run individual test script file of an application name in ReMoodWork:
    - `python manage.py test <app_name>. tests.<test_script>`

8. Recommendation for Enhancements

8.1 Achieved Current Tasks and Parts in ReMoodWork

The following parts and tasks were able to deliver in the project shown below:

✓ Completed setup for user registration and basic authentication process using Django, allowing users to identify as an Employee or Employer.

✓ Accomplishing two main features for gathering and producing records from Employee’s personal information:

  - Produced and fetched multiple pulse survey records for employees’ daily lifestyle activities.
  - Created and fetched a single meal planning assessment to measure employees’ health and dietary information.

- ✓ The system enables employers to select current employees at the same company and add food meal items to benefit their employees. The system also allows employees to choose and order a food meal item from restaurants and recipe links provided by their current employer.

- ✓ The system delivers an algorithm for filtering food items based on an employee's meal plan assessment/record.

- ✓ Wrote automated test scripts for verifying parts and component structure of ReMoodWork through unit testing (models and forms) and acceptance testing (views).

8.2 What implementation features and tasks are left and used for future work plans in this project?

There are some implementation features and tasks that were left in the final project and will potentially become used for future works used in this project due to limited time constraints shown below:

- ❖ ***Web Tokenization:*** Provides an essential security feature for enhancing and improving user authentication of ReMoodWork. Furthermore, this application uses a simple authentication process identifying users as employees and employers. While it achieves certain levels of security, it might not be enough to protect sensitive user information from potential vulnerabilities and exploits within the ReMoodWork application. However, implementing Web Tokenization can improve security by encrypting user data and preventing a few restrictive and unauthorized accesses in ReMoodWork. Web Tokenization can complement ReMoodWork’s secured application by generating unique tokens for each user upon authentication and utilizing these tokens to validate succeeding requests to the server. Overall, this component can be beneficial to accommodate and achieve higher security environments for users to store and access their data handled in ReMoodWork.

- ❖ ***Website deployment of ReMoodWork:*** Website deployment is a desired recommendation for developing ReMoodWork’s system. Deploying an application to a web server can provide extensive accessibility for all users, as well as scalability improvements, and allowing to become more reliable and efficient enough to optimize better performances of ReMoodWork. Additionally, a wide variety of web deployment options might become helpful in the future of this project, including cloud-based solutions like Amazon Web Services (AWS), Google Cloud Platform, and Microsoft Azure as potential candidates to deploy the website application of ReMoodWork.

- ❖ ***Django REST Framework:*** This feature provides an optimal suggestion for improving specific components and structures of ReMoodWork. While Django provides a great basis for developing web applications, it does not offer a built-in library for building and producing APIs or web APIS as part of a web interface used in this project. By transitioning to the Django REST framework, this application can enhance many functionalities and implementations, such as serializations, permissions/authentication, and viewsets/routers. This feature can improve ReMoodWork by refactoring and changing views to viewsets, updating URL routes, and integrating serializers used in the project. Overall, Django REST Framework shall provide robustness and scalability achievement for delivering future developments and enhancements of this software application.

- ***Third-party APIs/web APIs*:** Integrating third-party APIs/web APIs can be an effective way of enhancing the functionalities of ReMoodWork. Moreover, multiple third-party API/web API packages might help integrate this project in the future and can provide valuable features, such as API designs, documentation, and monitoring. Applying those packages can potentially improve user experiences, expand ReMoodWork's structured capabilities with Django's built-in functionality, and extend other services and data handled in this project.

- ***Incorporate some update and delete operations in ReMoodWork:*** Features (pulse survey, meal planning assessment, and ordering/removing food meal items) used in this project were mainly implemented and verified through creating and fetching data information of ReMoodWork only. However, there was no provided implementation on updating and deleting data information records for the provided features (pulse survey, meal planning assessment, and ordering/removing food meal items) used in the project due to limited time constraints. However, it would still be beneficial to incorporate those mechanics for future ideas used in this project.

- ***Cross-browser testing (Most likely Selenium):*** Cross-browser testing (such as Selenium) could be a potential fit for testing additional web page contents in ReMoodWork's software application and ensures that it can become compatible with various browsers and other operating systems happening in the future. While acceptance testing focuses on the application's functionality, such as HTTP requests, cross-browser testing delivers a broader scope of verifying user interactions with visual elements and layouts of ReMoodWork's application. By conducting and validating cross-browser testing, it should be able to identify and resolve any issues relative to the website's layout and design and make sure that the website displays the same content across all the web browsers and other appropriate devices. Cross-browser testing can also help improve user experiences of ReMoodWork. Therefore, integrating this testing method into the development process could significantly improve the quality and usage of ReMoodWork.

- ***Microservices Architecture and Integration with DevOps Environment:*** This project should plan to implement microservices architecture using Docker and integrate DevOps practices to enhance scalability and deployment process for ReMoodWork's application in the future. Furthermore, it decomposes into smaller independent services like API, database, and codebase that reduce broader complexities and provide rapid deployment for newer features, simple scalability, and increased fault tolerance to complement the infrastructures of ReMoodWork. Docker should also potentially deliver this project by containerizing each microservices and utilizing better deployment for expanding necessary environments of ReMoodWork that will be beneficial in the future. By integrating DevOps practices, it should ideally work with building and deploying microservices with Docker and potentially achieve the overall quality of ReMoodWork's application, such as continuous integration (CI) and continuous deployment (CD) handled in the future.

- ***(Optional, but highly recommended) Databases for handling high-traffic levels:*** The project uses SQLite (db.sqlite3) as a lightweight and serverless database. However, ReMoodWork might have some limitations working with SQLite (db.sqlite3) based on handling high-traffic levels.

Therefore, this project might migrate to a more scalable database as the application expands in the long run, such as PostgreSQL, MySQL/MariaDB, or MongoDB. Overall, having a database that handles high-traffic levels may have a firm advantage in improving the scalability and performance used in ReMoodWork.

- ***(Optional, but highly recommended) Alternative frontend frameworks for high-level visuals:*** The project uses HTML and some Bootstrap frameworks for its frontend portion. However, ReMoodWork might explore and accommodate other popular frontend frameworks to deliver high-level visuals and enhance user experience. Therefore, this project might be able to transition to other frontend frameworks for providing other pre-built user interfaces (UI), such as React, Angular, and Vue.js, that might integrate well with Django.

9. Conclusion to ReMoodWork as a software application product and the overall content of its development project

The issues of the Great Resignation have led to its impactful risks and challenging changes in the workforce conditions happening in the future. Furthermore, this trend has caused employees to resign and become dissatisfied with their jobs due to burnout as their mental health problem. As remote work continues to be a popular choice for several companies to accommodate in this modern age of employment services, it has increasingly become more difficult to address mental health challenges for all employees doing their tasks and activities remotely. By making the remote work environment much more valuable in tracking employees' mental health status, ReMoodWork will deliver this concern by supporting employees to achieve better mental health and increasing company culture for a better reputation and investments in potentially growing its software application. The characteristics and functionalities of ReMoodWork provide its unique software platform seeking to help employees control their mental health while working remotely. In addition to this software application, it covers pulse and meal planning (assessment) surveys as the base features of this application.

This software application takes on delivering its evaluation and decisions based on employees' daily lifestyle activities to work remotely for its pulse survey as the first feature of its development. In addition to its implementation logic of pulse surveys, it measures employee engagement based on their mental health and daily experiences to let organizations monitor and improve employees' lifestyles to work anywhere remotely. The pulse survey feature of ReMoodWork also incorporates emotional pictograms of an employee that represent Emojis on determining senses relating to human emotional behavior to reduce job burnout for each remote worker.

ReMoodWork also comprises its development and implementation of meal planning (assessment) surveys by assisting employees in exploring and finding a proper food diet based on evaluating their nutritional needs based on food allergies, dietary goals, and recommended meals to fit with requirements recorded by the employee's health data information.

In conclusion, ReMoodWork provides a helpful software application tool to change the company's culture by providing healthy lifestyle choices and communicating effectively between employees and employers working remotely. The software application platform will become a potential tool for achieving daily progress and a productivity goal of an employee's mental health with the small steps of developing pulse and meal planning (assessment) surveys as a driving force of this developed project.

The atmosphere of remote work will likely become more common as the Great Resignation continues to affect jobs in the future. Therefore, by improving certain conditions of a remote work environment with ReMoodWork, employers can accommodate specific resources for employees to cope with their mental health and provide comfort in producing a much more positive and productive working structure of their job. By utilizing ReMoodWork, companies can create a better remote work culture that benefits employees' mental health, leading to higher retention rates and better employee satisfaction and business performance for the future.

10. Bibliography (References)

[1] Ellerbeck, Stefan. (2022, June 24). *The Great Resignation is not over: A fifth of workers plan to quit in 2022*. World Economic Forum, 24 June 2022, https://www.weforum.org/agenda/2022/06/the-great-resignation-is-not-over/.

[2] *Avoiding Stress & Burnout*. Wrike, https://www.wrike.com/remote-work-guide/remote-work-burnout-mental-health/.

[3] Lalwani, Puja. (2021, March 11). *What Is Employee Engagement? Definition, Measurement, Drivers, and Strategies, with Examples*. Spiceworks, 11 March 2021, https://www.spiceworks.com/hr/engagement-retention/articles/what-is-employee-engagement/.

[4] Lu, Xuan, et al. *Emojis Predict Dropouts of Remote Workers: An Empirical Study of Emoji Usage on GitHub*. PLOS ONE, 26, January 2022, pp. 1–21, www.proquest.com/docview/2488775506?accountid=9840&parentSessionId=JcRvzD3mACG6Qfupha%2BzgCMzEKExZRzwCGgvwln2tXc%3D.

[5] Oh, Hyungik, and Ramesh Jain. *From Multimedia Logs to Personal Chronicles*. Association for Computing Machinery, 23-27, October 2017, pp. 881-889, https://dl-acm-org.lib-proxy.fullerton.edu/doi/10.1145/3123266.3123375.

[6] Jain, Ramesh, et al. *A Survey on Food Computing*. ACM Computing Surveys, September 2019, 52(5), pp. 1-36, https://dl-acm-org.lib-proxy.fullerton.edu/doi/10.1145/3329168.

[7] *Customizing authentication in Django*. Django, https://docs.djangoproject.com/en/4.1/topics/auth/customizing/#django.contrib.auth.models.AbstractBaseUser

[8] *Velocity Chart*. Agile-Scrum, https://www.agile-scrum.be/whats-great-scrum-methodology/velocity-chart/

[9] *Difference between MVC and MVT design patterns*. GeeksforGeeks, 14 September 2022, https://www.geeksforgeeks.org/difference-between-mvc-and-mvt-design-patterns/

[10] *Working with forms*. Django, https://docs.djangoproject.com/en/4.1/topics/forms/

[11] *Introduction to JSON Web Tokens*. Django, https://jwt.io/introduction

[12] *What is Tokenization?* TOKENEX, https://www.tokenex.com/blog/what-is-tokenization/

[13] *Cloud computing with AWS*. AWS, https://aws.amazon.com/what-is-aws/

[14] Django REST Framework, https://www.django-rest-framework.org/

[15] Swagger, https://swagger.io/

[16] Selenium, https://www.selenium.dev/

[17] Ostrowski, Radek. *Getting Started With Docker: Simplifying DevOps*. Toptal, https://www.toptal.com/devops/getting-started-with-docker-simplifying-devops

[18] PostgreSQL, https://www.postgresql.org/about/

[19] *PostgreSQL 8.2.23 Documentation*. PostgreSQL, https://www.postgresql.org/docs/8.2/high-availability.html

[20] Dhaduk, Hiren. *Best Frontend Frameworks for Web Development in 2023*. Simform, November 28, 2022, https://www.simform.com/blog/best-frontend-frameworks/

[21] Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others. Altexsoft, October 27, 2021, https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/