Homework 5

Mohammed Algadhib
algadhim@oregonstate.edu
Johnny Chan
chanjoh@oregonstate.edu

Riley Kraft
kraftme@oregonstate.edu

Diego Saer
saerd@oregonstate.edu

1. USER STORIES

A. DAILY HEALTH QUESTIONS

As a user, I want to make note of how I'm feeling overall today, so that anything potentially concerning can be recognized early.

- Questions on scale from 0 to 10 are generated based on the medications taken by the patient.
- The answers are then compared with the side effects and symptoms of the medications and based on that it will be determined by the system if there is anything potentially concerning.
- If there is any concern a notification will be created to notify the user.

B. DAILY SYMPTOMS SURVEY

As a user, I want to make note of all the symptoms I'm feeling today, and where, so that anything potentially important about my health does not go unnoticed.

- An interactive body diagram is presented to the user. The user can select a part of the body to enter symptoms felt there.
- The answers are then compared with the symptoms database to check if anything is important.
- Notifications will be generated if there are any unusual symptoms.

C. DECIPHER PRESCRIPTION DATA

As a user, I want to be able to take a picture of my/the patient's prescription label and have it fill in all relevant information for my medication, so that I don't have to spend so much time manually entering all the data.

- Patient or caretaker uses camera on mobile device to capture images of a prescription label
- Photos are sent to an Optical Character Recognition (OCR) API
- JSON returned to client with deciphered text and parsed into respective fields

D. MEDICATION DOSE REMINDER

As a user, I want to be alerted when I need to take a medication dose or have an upcoming doctor's appointment, so that I can stay on top of my/the patient's personal healthcare.

- Medicine schedule is imported when adding medication either by photograph import or manual entry.
- Schedule and alerts for the medicine can be modified within the medication.
- When an alert is triggered, a notification will appear for the user along with the option to vibrate the phone or make a sound.

E. RECORD DOCTOR INFORMATION

As a user, I want to record my doctors' contact information, so that it's easier to get in touch with them when I have an appointment or an emergency.

- Doctor's info is imported by taking picture of medication
- Additional Information can be added manually while adding medication.
- Doctor's information can be added manually by selecting the appropriate icon from the homepage.
- Doctor's info is stored locally and on the external database.

F. ALERTS REPOSITORY

As a user, I want to review cautionary alerts at any time so that I don't dismiss anything that might be important.

- User selects Alerts icon from the homepage, which takes them to a chronological list of notifications issued in the past 30 days.
- Notifications are color coded based on the level of potential severity.
- User can select a notification, which populates a window with the details of the notification.

G. RECORD IMAGE OF MEDICATION

As a patient, I want to take pictures of my different medications, so that it's easier to identify them in my pill box when it's time to take a dose.

- Patient/caretaker uses camera on mobile device to capture image of medication OR uploaded from a file to the web application
- Image is stored and retrieved by application on user's client device

H. VIEW MEDICATION INSTRUCTIONS

As a patient, I want to get instructions with my medication reminders in case I forget or don't have the medication bottle in front of me.

- Medicine instructions can be imported when creating a medication by either photograph or manual import.
- Medicine instructions can be modified thereafter by modifying the medication.
- The instructions stored will appear in the medicine dose reminders.

I. GENERAL SYMPTOMS RECORDING

As a patient, I want to record symptoms I'm feeling that may not be associated with my medications so that my caretaker and doctors are aware of all my health concerns.

- An interactive body diagram will be presented to the user. The user will be able to click on any body part and select the symptoms that the patient is feeling there.
- The data will be recorded and then formatted properly to be included in the health report.

J. SET DOSAGE REMINDERS

As a user, I want to set reminders in my calendar for taking my medications so that I don't forget and miss a dose.

- Patient/caretaker selects how many doses a day they need to take
- For each dose in a day, the user defines a time of day for each dose
- Reminders are saved to the calendar

K. MANUALLY ENTER MEDICATION

As a user, I want to be able to manually record/edit my medications and supplements so that I don't always have to rely on using a camera.

- User selects the Add Prescription or Add Over the Counter icons from the homepage to be taken to a new medication form.
- User is able to manually enter all available data for a prescription or over the counter medication/supplement.
- System does not need to access OCR API.

L. USER FRIENDLINESS

As an elderly patient, I want to be able to see and/or hear the app information clearly so that I don't have to struggle while using it.

- The information (ie. text and pictures) are presented clearly in a visible size and colors.
- Text to audio feature is integrated to the user interface.

M. PRESCRIPTION REFILL REMINDER

As a user, I want to get a reminder when I'm running low on my prescription, so that I don't run out of my medication.

- When importing medication a reminder can be manually added so that an alert appears when a specified number of doses remain.
- After import, by viewing the medication this reminder can be modified as well.
- The system will keep track of the dosage left by taking the total doses in the prescription and subtracting one dose with each time the patient takes their medicine.
- Once the remaining doses reach the amount specified by the patient, the patient receives a notification.

N. ACCOUNT SECURITY

As a user, I want to secure my account so that I can keep my private data safe.

- It will be required to create a strong password when creating an account.
- The account credential will be encrypted and stored in a secure database to insure that they are going to be protected against attacks.

O. PROFILE MANAGEMENT

As a user, I want to be able to edit my profile so that I have the most up to date data.

- There will be a user profile icon on the homepage.
- A drop-down menu will have settings on it.
- Once patient taps and the user is on the settings page, user can edit account details.
- A window pop-up asking user for final confirmation.
- A large save button is on the bottom of the page.

P. VIEW ALL MEDICATIONS

As a user, I want to see a list of all my/the patient's medications so that I can ensure I have the most up to date data.

- On the homepage/homescreen there will be an icon called "Medications" which will contain the list of medications currently being taken by the patient.
- The patient can press on the icon to bring them to a page that lists all the medications currently being taken along with the option to update or delete any.

Q. GENERATING REPORTS

As a user, I want to send all recent information about my medications and overall health to my doctor, so that I don't forget any important information during my next doctor's appointment.

- User select Reports icon on homepage to take them to the list of generated reports, in chronological order.
- User can choose to select an existing report, or generate a new one.
- User select report(s) from the list to review, then submit to a doctor via print or email.

R. INVITE CARETAKER

As a patient, I want to designate a caretaker to access certain parts of my account so that I have help managing my healthcare.

- Patient invites caretaker by email to create/merge their own account with the patient's account
- Patient sets permissions for what the caretaker can access in their account
- Caretaker can add, delete, and manipulate data for which they have access on the patient's account

S. ACCESSIBILITY

As a user, I want to access my account from any type of mobile or browser interface so that I don't have to be concerned with restricted compatible hardware.

- User can access their account, data and system capabilities from any mobile or web browser client.
- The interfaces for each type of client are designed for high usability in terms of elderly users.
- There is an added security measure which tracks the devices the user has used to access their account.
- If a previously unknown devices sends a request to login, the system sends an email to the account holder to confirm the new device.

T. SEAMLESS CONNECTIVITY

As a patient, I want to be able to access my medication information at any time in the event I'm not feeling well or end up in the hospital.

- Patient/caretaker, if connected to the internet, can always access their data
- Patient.caretaker, if not connected to the internet, can always access the most up to date data since losing connection

2. TASKS

NOTE: All estimated times include testing.

A. DAILY HEALTH QUESTIONS

This Story is one of the essential use cases for this system, as a result it is a high priority. However, one main challenge when implementing this story is generating accurate questions to precisely detect alerting side effects. This story is dependent on creating the side effects and the survey classes in the database. Without side effects data, this story uses default questions, making this a medium-low cost for the group. The addition of a question generating algorithm makes this a high cost story for the group.

1. Setup database for survey class and to store questions and answers - must come first since we need a place to derive questions from and store answers to
2. Write default questions and add as records in Question table.
3. Create Take Survey icon on homepage
4. Create web page to display user instructions
5. Create web page user interface to answer survey questions - medium.
6. Write code to take user from homepage to survey page
7. Write code to retrieve questions from the database.
8. Write code to present questions one at a time.
9. Write code to record answers to the database.
10. Create confirmation web page
11. Write code to query severity of answer.
12. Write code to create Notifications based on severities.
13. Write code to take user to Alerts repository webpage.
14. Write question generating algorithm based on current medications and known side effects - this is of the lowest priority for this user story since is relies on data generated by the user, hence it can be implemented later.

Estimated Time: 15 hours for tasks 1-10; +8 hours for tasks 11-13, and is dependent on Medications DB API and SideEffects class; +8 hours for algorithm, and is dependent on Medications DB API and SideEffects class.

Due: Thursday, August 9th

B. DAILY SYMPTOMS SURVEY

This story is considered to be high priority as it is one of the main functionalities of the system. Though this is dependant on creating databases to store the symptoms and daily surveys. This story is also of a high cost considering the available symptoms rely on side effects added by the Medications DB API. This team does not have much experience working with outsourced API's.

1. Setup database to record symptoms associated with surveys - must come first since we need a place to store the user's symptom data
2. Create webpage with interactive human body diagram - priority
3. Write code to retrieve side effects from database - priority
4. Write code to temporarily save selected symptoms - priority
5. Write code to record selected symptoms in database - priority

6. Write code to query severities of symptoms recorded - priority
7. Write code to create Notifications based on returned severities - priority
8. Write code to take user to Alerts repository webpage

Estimated Time: 40 hours

Due: not slated to be due in the next 2 weeks

C. DECIPHER PRESCRIPTION DATA

As one of the major requirements since the inception of the project, deciphering images of prescription labels into text data is a high priority. However, the lack of experience of the software engineering team in working with outsourced API's also makes this task a high cost.

1. Setup database to store prescription label data - this task must come first since we need a place to store the data after it's been retrieved
2. Setup mobile client to access camera feature - this comes next since we can't call to the API without having pictures of the prescription labels
3. Setup access to storage on client device to hold and reference images
4. Write code to POST images to OCR API
5. Write code to parse JSON text received from OCR API, and place in respective application fields
6. Write code to store data to database

Estimated Time: 40 hours

Due: not slated to be due in the next 2 weeks

D. MEDICATION DOSE REMINDER

The priority is medium for this story. Although one of the important functions of the system is to help remind patients to take their medications, this functionality is dependent on multiple components working together, e.g. the medicine, notification, and calendar objects. The creation of a GUI for the notifications screen adds significant time to this project since the team lacks experience in this field. This project is itself high cost to develop even without the GUI.

1. Design a table to store medication use information in the external database. This should be done first if the connection to the external database is working to aid in determining the system is working correctly.
2. Implement a GUI for notifications. This will aid in testing if the system is working correctly and be part of the final product. This may be implemented later.
3. Implement the notifications object. So that all objects that require notifications can be tested. Also make sure that a medication use notification gets sent to the external database and stored in the table designed previously.
4. Implement the event object so that the calendar object can use it to function correctly.
5. Implement the calendar object that uses the event object to keep track of the events. The calendar object must also be able to send a signal to the notification object to create a new notification. This must be implemented before the medication object as the medication object will call the calendar object to create a new set of events.
6. Implement the medication object making sure the member functions can call on the calendar object to create and modify events.
7. Make sure all components above work well with each other.

Estimated Time: 20 hours for GUI; +35 hours if dependent tasks are not already completed

Due: not slated to be due in the next 2 weeks

E. RECORD DOCTOR INFORMATION

This is a low priority functionality as it serves mainly as a contact list which is not the primary purpose of the system. Also, with the exception of the prescription and report classes, the objects containing the doctor's information are stand alone and don't interact with the rest of the system. The creation of a GUI adds significant time to this project since the team lacks experience in this field. For the group, this story is a high cost with a GUI and a medium cost without one.

1. If the prescription and report classes are to be developed first, place a stub anywhere the doctor class is required.
2. If a connection to the external database has been established, design a table which holds the doctor information. This will aid in determining if the doctor class is functioning correctly.
3. Implement a GUI for the form where the doctor's information is added. This will also aid in determining if the class is functioning correctly and be used in the final product. This may be implemented later.
4. Implement the address class which stores address information for any doctor added. It is logical to build this before the doctor class as the doctor class uses the address class for some of its functions.
5. Implement the doctor class which points to an address object and stores the doctor's contact information in its member variables.
6. If the prescription and report classes were already made, remove the stubs and connect these classes' functionality to the doctor class. Otherwise stubs for these classes will have to be made.
7. Make sure that the doctor class is working well with the objects it interacts with and that upon creation, modification, or deletion the external database is being updated.

Estimated Time: 10 hours

Due: Thursday, August 16th (provisional)

F. ALERTS REPOSITORY

High Priority. The alert system has always been a high priority item since the project's inception. A patient may receive an alert but may not be able to address the issue at the moment so a repository for alerts is crucial. The entire project is high cost but if the GUI is deferred until later it is medium cost.

1. If a connection to an external database is already implemented, design a table which holds data relevant to storing alerts. This is an eventual requirement and helps test the system.
2. If a GUI to the alerts repository is not created. It may be created at this stage to aid with testing. Otherwise this may be implemented later.
3. Create the notifications object which will take certain information such as id, type, date/time , severity, and description.
4. A data structure holding all the notifications and is able to add and archive them must be implemented to keep track of all the notifications locally.

Estimated Time: 10 hours without GUI; +25 hours with GUI (mobile devices only)

Due: Thursday, August 16th (provisional)

G. RECORD IMAGE OF MEDICATION

Recording an image of each medication has been a requirement since the inception of the project, however this piece of data is not as vital as the other data required for recording a new medication. Thus, this story has a medium priority. The simplicity of the tasks for this story paired with the lack of knowledge of mobile development would render a medium cost for this software team.

1. Setup storage for pictures on client device - must come first so the application knows where to store and access the images uploaded
2. Setup mobile client to access camera feature OR create webpage and write code to upload image - must come next to obtain images
3. Write code to store images on client device for later reference

Estimated Time: 10 hours

Due: not slated to be due in the next 2 weeks

H. VIEW MEDICATION INSTRUCTIONS

Medium Priority. This feature essentially adds convenience to the user but not much else. It can be important if the user misplaces their original directions though. Since this story is a logical next step from the medication dose reminder, it would be a low cost to develop after the reminder is already in place. Otherwise it would be a high cost to develop from the ground up.

1. Make sure the the notifications to use medications are properly set up (55 hour estimated task).
2. Modify the notification object's call to create a medicine use notification. Allow it to hold an additional parameter that holds the directions.
3. Make sure the notification created displays these directions.
4. Now that the notification object has the extra parameter for receiving the a medicine's instructions, modify the medicine object's call to the notification object.

Estimated Time: 10 hours (on top of a 55 hour task)

Due: not slated to be due in the next 2 weeks

I. GENERAL SYMPTOMS RECORDING

This story is dependent on story B, and if story B is completed, with minimal modifications, this story will be completed as it requires less complications that story B. Hence, this story is of a low priority and low cost to the group.

1. Add feature to webpage for user's ability to add new symptom - priority
2. Create webpage/window for adding details of symptom - priority
3. Write code to record symptom in database - priority

Estimated Time: 10 hours

Due: not slated to be due in the next 2 weeks

J. SET DOSAGE REMINDERS

This feature is optional within the software application, though the calendar cannot send alerts to the user regarding taking their medication, thus this story has a medium priority. The tasks and skills required of implementing this story render it of a low cost.

1. Setup database to store reminder data - this is the first task to accomplish in order to test the proper recording and retrieving of the data
2. Create webpage/mobile interface to input reminder data - must be accomplished next in order to get the data that needs to be recorded
3. Write code to record data to database - next logical task to save the user's data entered

Estimated Time: 10 hours

Due: Thursday, August 16th (provisional)

K. MANUALLY ENTER MEDICATIONS

High priority. This is a convenience for the user when user's mobile camera is not functioning, or the user is utilizing a web browser.

1. Setup database to record medication objects - priority
2. Create Add Prescription and Add Over the Counter icon on homepage - priority
3. Create medication form webpage - priority
4. Write code to take user to medication form from homepage - priority
5. Write code for web app to connect with database - priority
6. Write code to record data to database - priority
7. Design a window for the confirmation from the user.

Estimated Time: 15 hours

Due: Thursday, August 9th

L. USER FRIENDLINESS

One of the most important aspects to this system is usability, thus, this story is high priority. This story is mostly doing research to find the key design elements that makes the usability of the system more friendly based on the targeted audience, then making sure to implement the usability features when creating each user interface. The simplicity of gathering this information makes this a low cost story for the group.

1. Find the optimal color scheme and font size - priority
2. Create homepage as template for remaining webpages - priority
3. Integrate findings into design of each webpage/mobile interface - priority
4. Find a tool that reads text to audio- priority
5. Add audio tool to instruction webpages - priority

Estimated Time: 4 hours for tasks 1-3; +6 hours for remaining tasks

Due: Thursday, August 9th

M. PRESCRIPTION REFILL REMINDER

Low Priority. This is a convenience as the user or the caretaker can visually inspect how much medication is left. The creation of a GUI adds significant time to this project since the team lacks experience in this field. Creating this from the ground up will be a significant task for the group.

1. If a connection to the external database has been set up, design a table to store medication refill information in it. This will aid in determining if everything is working correctly and is required in the final product.
2. Implement a GUI for notifications. This also is a requirement for the final product but can be useful in determining if the story is working correctly. This may be implemented later.
3. Implement the notifications object so that all objects that require notifications can be tested.
4. If applicable, make sure that a medication refill notification gets sent to the external database and stored in the table designed previously.
5. Now that all the features that help display and store the notification are set up, implement the event object so that the calendar object can use it to function correctly.
6. Implement the calendar object that uses the event object to keep track of the events. The calendar object must also be able to send a signal to the notification object to create a new notification and must be able to send a signal to the medication object to update the medication quantity.
7. Implement the medication object making sure the member functions can call on the calendar object to create and modify events.
8. Make sure that when called upon by the calendar object, the medication object can update the medication quantity.
9. Make sure that the medication quantity falls to or below an amount set by the user, a call to the notification object is made
10. Make sure all components above work well with each other.

Estimated Time: 10 hours without GUI; +25 hours to create GUI

Due: not slated to be due in the next 2 weeks

N. ACCOUNT SECURITY

This story is important toward the final versions, however, in development and testing it is not critical. As a result, this is a medium priority story.

1. Find authentication framework to be integrated to this system - research must be done first to ensure we use the proper resources
2. Integrate the selected authentication framework to the system - must be integrated next before any pages can be tested
3. Create the user interface for the login/sign up pages - implemented last to ensure good testing

Estimated Time: 10 hours; more if unfamiliar with authentication framework

Due: not slated to be due in the next 2 weeks

O. PROFILE MANAGEMENT

This is a high priority story as proper profile management is essential to any application that handles user accounts and personal data. It also establishes the connection to the external database which can be used on other stories. This is a high cost story though. The GUI adds much more work since the team doesn't have much experience making them. Even without the GUI, getting the database set up and the syncing functionality working adds significant effort.

1. Establish a connection to a secure external database. This is crucial to some of the most basic profile management functionality.
2. Design and create a table which handles user information such as ID, username, password, account type, permissions, and companion ID.
3. If it desired at this stage, create a GUI that contains the login screen. This is an eventual requirement and can help testing the entire profile management process.
4. Implement the account class and its two subclasses, patient and caretaker.
5. Ensure the account class can communicate with the external database and has only access to the user's pertinent information.
6. Ensure that after logging in, the user's local storage syncs with the external database correctly.

Estimated Time: 15 hours; +25 hours for GUI and local database (mobile device only)

Due: not slated to be due in the next 2 weeks

P. VIEW ALL MEDICATIONS

Low priority. This can be implemented more easily once the rest of the system is in place. The creation of a GUI adds significant time to this project since the team lacks experience in this field and leads to a high cost. It would be a medium cost without the GUI.

1. Design a table to store medication information in the external database.
2. Implement a GUI to display medications. This may be implemented later.
3. Write code to retrieve medication objects from the database.
4. Make sure that upon medication creation/update/deletion the external database is updated.
5. Make sure all components above work well with each other.

Estimated Time: 15 hours without GUI; +25 hours with GUI (mobile device only)

Due: Thursday, August 16th (provisional), not including task P2

Q. GENERATING REPORTS

Low priority. Even though eventually this will be required. This story relies on other systems being in place so that it can pull information from them and create the report. Not recommended to be done immediately. There are many elements essential to report generation that the team does not have experience with, namely the GUI, sending documents to a printer, creating pdfs, and emailing information. Therefore, this is a very high effort task.

1. First the material the report generator uses to pull information must be set up. If the external database is not set up for this then try the local storage. If the local storage is not set up for this then create stubs for doctors, surveys, medicines, notifications, etc.
2. If desired at this stage, set up a GUI for the report generation.
3. Implement the report class.
4. Implement the pdf view subclass of the report class.
5. If access to remote storage is currently enabled, test that the report gets sent there after creation. Otherwise set up a local storage area for the reports and ensure they are stored there.
6. Ensure that the email and print functionality are working.

Estimated Time: 15 hours without GUI; +25 hours for GUI (mobile devices only)

Due: not slated to be due in the next 2 weeks

R. INVITE CARETAKER

It is entirely optional for a patient to invite a caretaker to help them manage their account, and a caretaker would not have any capabilities that a patient account would not have. Thus, this story is of a low priority. The tasks and skills required to implement this story render it a low-medium cost.

1. Setup database to record companion ID in Account class and record permissions - must occur first in order to test the capabilities of the code to record and retrieve relevant data
2. Create webpage/mobile interface to invite a caretaker and set their permissions - patient must invite caretaker first
3. Create webpage for caretaker to accept invitation and create their own account - caretaker then accepts, or declines, invitation
4. Write code to record companion ID's and permission in database - relevant fields are then set in the database for reference

Estimated Time: 20 hours

Due: not slated to be due in the next 2 weeks

S. ACCESSIBILITY

This is currently a low priority as it is just a convenience to the user. This user story does not have bearing on the overall system functionality concepts. Though, the code and skill needed to support the system on multiple types of clients makes this a high cost story for out team, as many of us do not have experience with mobile development.

1. Create webpage interfaces for initial user access and testing - priority
2. Create mobile client interfaces for final user access and testing - priority
3. Setup database to store user's recognized devices for security reasons - priority
4. Design an email that generates confirmation passcode and instructions for user to approve new device - priority
5. Create interface to accept passcode and record hardware ID as approved by the user - priority

Estimated Time: 120+ hours, including the time it takes to make all the web and mobile interfaces

Due: not slated to be due in thee next 2 weeks; this is an ongoing user story that is dependent on the completion of other user stories

T. SEAMLESS CONNECTIVITY

To ensure the reliability of the software, this user story is of a high priority. However, the task requirements and lack of current skills of the software engineering team in creating mobile applications also renders it a high cost.

1. Setup database to record all data - must come first in order to provide the outline for the local database
2. Set up local database on mobile client to record all data - comes next in order to record mobile client data
3. Write code to synchronize local database and database server, regularly - comes next in order to test the compatibility of the local database and database server, as well as the seamless updates between the two modules

Estimated Time: 40+ hours

Due: not slated to be due in the next 2 week

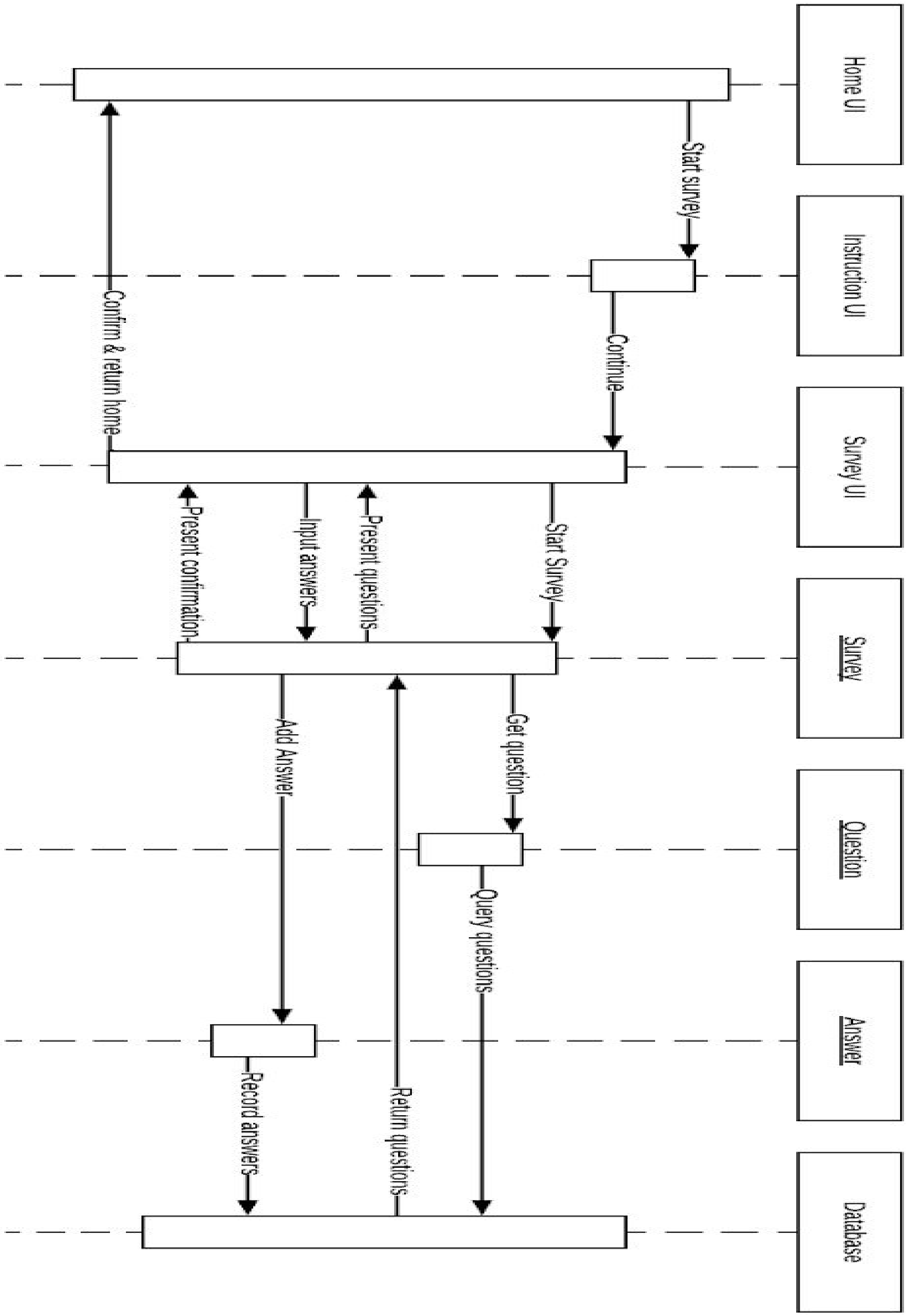
3. PLAN OUTLINE

WHEN	WHAT	WHO
Monday	L1 - research usability features of color, font size, etc. (~1 hour) L2 - Create homepage (~2 hours) Design and perform tests	Diego & Mohammed
	K1 - setup Medications table in database (~1 hour)	Riley & Johnny
	A1 - setup Survey, Question, and Answer tables in database (~1 hour) A2 - write default questions and add as records to Question table (~1 hour)	Diego & Mohammed
Tuesday	K2 - create Add Prescription and Add Over the Counter icons to homepage (~1 hour) K3 - create medication form webpage (~4 hours) L3 - integrate usability features during development Design and perform tests	Riley & Johnny
	A3 - create Take Survey icon on homepage (~30 min) A4 - create webpage for user instructions (~30 min) A5 - create webpage to answer survey questions (~2 hours) L3 - integrate usability features during development Design and perform tests	Diego & Mohammed
Wednesday	K4 - write code to take user to medication form from homepage (~30 min) K5 - write code for webpage to connect to database (~30 min) K6 - write code to submit data in fields to database and (~4 hours) K7 - create confirmation webpage (~30 min) Design and perform tests	Riley & Johnny
	A6 - write code for icon to take user to survey page (~30 min) A7 - write code to retrieve questions from database (~1 hour) A8 - write code to present questions one at a time (~2 hours) A9 - write code to record answers in database (~2 hours) A10 - create confirmation webpage (~30 min) Design and perform tests	Diego & Mohammed
Thursday	K - final testing (~2 hours)	Riley & Johnny
	A1-10: final testing (~2 hours)	Diego & Mohammed
	Present stories to customer for evaluation and revisions	Team and Customer

4. SPIKES & SEQUENCE DIAGRAMS

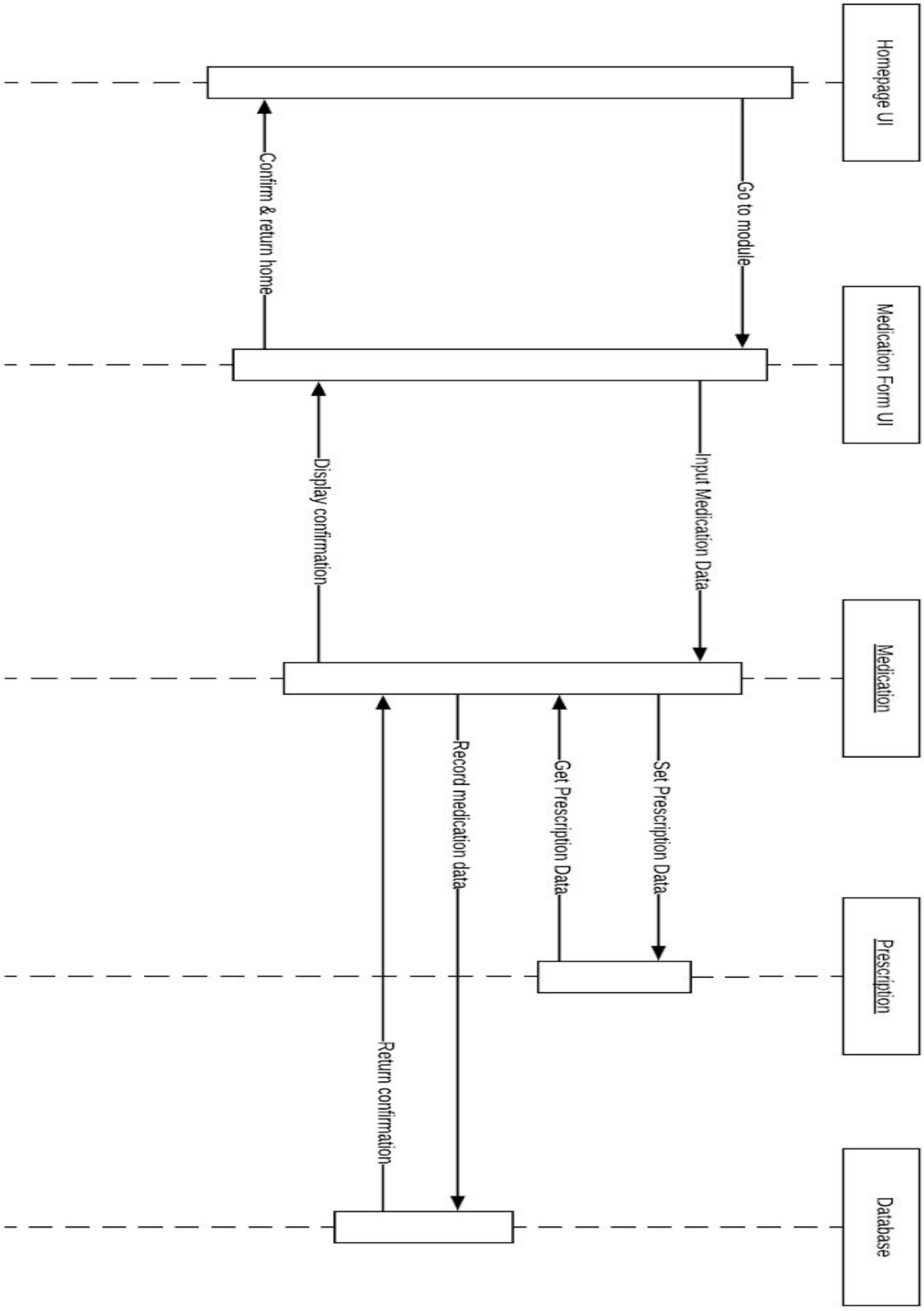
4.1 USER STORY A, TASKS 1-10

The following is a sequence diagram depicting the flow of events for our system between user interfaces, class methods and database storage for user story A. The diagram is limited to the tasks scheduled to be completed in the upcoming week.



4.2 USER STORY K, TASKS 1-7

The following is a sequence diagram depicting the flow of events for our system between user interfaces, class methods and database storage for user story K. The diagram is limited to the tasks scheduled to be completed in the upcoming week.



4.3 USER STORY L, TASKS 1-5

The following is a write-up for the spike performed for this user story. The team's lack of knowledge/experience of usability makes this an appropriate, and high priority, user story for this project. The following is the knowledge gained by researching and testing the best usability features for our user interfaces.

1. Find the optimal color scheme and font size.

   - Font must be at least 12 pt and there must be an option to resize.

   - All colours used must be vivid and neighbouring colours, like in adjacent buttons, must contrast with each other.

2. Create homepage as template for remaining webpages.

   - All pages, including the homepage, must have ample whitespace.

   - There must be plenty of space between lines and paragraphs.

   - All headings must be distinguished by size and colour.

   - There must be no horizontal scrolling and minimum use of vertical scrolling.

3. Integrate findings into design of each webpage/mobile interface.

   - Half the design of this system is focused on mobile device access, thus much of the usability features need to be attributed to small screen interfaces.

   - Make sure all buttons are coloured and then highlighted when moused over or pressed.

   - When filling out long forms, remind the user of what has been completed, e.g. change the section from grey to green and say completed.

   - Make error messages clear. Also make sure the user can easily find the error when filling forms (colour it), and knows what format to use when typing in information by giving examples or other means.

   - Don't make major overhauls to the website design in the future as many seniors create written instructions for how to use websites.

   - Prompts must make use of action verbs, e.g. click here to return to survey.

4. Find a tool that reads text to audio.

   - An example would be NaturalReader.

   - It may also be worthwhile to pay a professional to record these the instructions and send team the audio files to reduce the chances of the text to speech reader making a mistake.

5. Add audio tool to instruction webpages.

   - Identify parts of the program that may need text read aloud and insert a button next to the text that reads it.

5. REFERENCES

[1] Agile Modeling. *User Stories: An Agile Introduction.* [Online] http://www.agilemodeling.com/artifacts/userStory.htm

[2] Campbell. Feb 2015. *Designing For The Elderly: Ways Older People Use Digital Technology Differently*. [Online] https://www.smashingmagazine.com/2015/02/designing-digital-technology-for-the-elderly/

[3] Dailey. *How Senior Friendly is your website*. [Online] https://www.illuminage.com/files/2014/05/webinar-050913.pdf

[4] Moth. May 2013. *Six design tips for making your website senior friendly*. [Online] https://econsultancy.com/blog/62815-six-design-tips-for-making-your-website-senior-friendly

[5] Mountain Goat Software. *User Stories.* [Online] https://www.mountaingoatsoftware.com/agile/user-stories

[6] NaturalReader. [Online] https://www.naturalreaders.com/index.html