Lab 1 – Thought Locker

Deshawn Bailey

Old Dominion University

CS411

Professor James Brunelle

15 September 2023

Version 2
LAB 1 – THOUGHT LOCKER

Table of Contents

1 Introduction 4

2 Product Description 5

2.1 Key Product Features and Capabilities Error! Bookmark not defined.

2.2 Major Components (Hardware/Software) 5

3 Identification of Case Study 7

4 Product Prototype Description 9

4.1 Prototype Architecture (Hardware/Software) 9

4.2 Prototype Features and Capabilities 9

4.3 Prototype Development Challenges 9

5 Glossary 10

6 References 11
List of Figures

Figure 1: Major Functional Component Diagram 6
Figure 2: Prototype Major Functional Components Diagram 10

List of Tables

Table 1: Thought Locker RWP v. Prototype Features 10
Dementia is a very unfortunate neurological disorder that effects the patients mental remarkably. There are different levels to the disorder’s severity which I will label mild/moderate and severe. Approximately one in ten Americans aged 70 and older are affected by some form of dementia. The severity of dementia varies, with 50.4% of cases classified as mild, 30.3% as moderate, and 19.3% as severe. Some symptoms of mild to moderate dementia can include causes the patient to lose memory of recent things such as item placement and many other small actions that occur in the day which can also lead to many mood changes in the patient (Alzheimer’s Society, n.d). Although the patient experiences the symptoms, they typically prefer to keep some form of independence instead of depending on a caregiver or family member to do every little task for them. A severe case of dementia include more drastic symptoms like very drastic memory loss where the patient can forget important relatives as a whole, loss of communication and in some cases hallucinations or delusions. In the case of severe dementia, the patient typically relies on a caregiver which is someone who helps with daily tasks of the dementia patient. It is no secret that dementia is a living nightmare for an elder or relative of someone with dementia and the disease steadily increases over the years. The disorder or long term things that has happened to them, constantly experience changes in attitude or their mood, and it also makes it very difficult to plan task or remember many of the tasks they have planned for themselves. With that in mind, this will lead to the dementia patient needing many reminders or a specific person to aid them in daily tasks such as bathing, eating, medicine intake and so on.
2 Product Description

Thought Locker is a mobile application that is used to help dementia patients maintain their independence through many of the applications features. Thought Locker includes features that help the patient keep up with their schedule, location of different items and contacts of different caregivers and family members. In addition to aiding the patients, the application also provides features that help caregivers with monitoring the patient movements, monitoring the patients tasks (rather daily or scheduled) and the location of the patients items.

2.1 key features

Thought Locker is a mobile application that works across both Android and IOS devices. The applications has two primary roles that are “Caregiver” and “Patient”. To start, both roles will have access the calendar feature. The calendar feature includes the ability to set reminders for certain events. The calendar of the patient role is accessible through the caregiver role allowing the caregiver to view or edit different tasks and upcoming events. The Item registration feature allows for both roles to add different items under the role of the patient. Items that are registered can also be tracked so that it is able to be found in any case of misplacement. The patient role will also have access to a Contact center that stores emergency contacts that can be family members, loved ones or emergency services like the police fire department and a selection of caregivers.

Event Viewer is a caregiver role specific feature that is included in the application that allows the caregiver to keep track of other caregiver roles tasks as well as the patients condition. The tasks can be set with alerts that range from low, medium, and high which further informs the caregiver on the importance of certain tasks that were completed or uncompleted. The caregiver
role will also have the ability to register and delete sensors in the application. The sensors can detect different things like motion and unusual temperatures. The caregiver role can also make reports about different tasks and unusual activity that may happen like a stove being left on or a door being opened at an unusual time of night. Remote control is another feature that is accessible through the caregiver role. This feature allows for the caregiver role to access the patient role’s user interface remotely which means the caregiver role can take control of whatever the patient role is currently doing. The caregiver role will basically be the manager of the patient role features and accessibility.

**Table 1**

*Major Functional Components*

![Diagram of Major Functional Components]

2.2 Major Components (hardware and software)

The software utilized in creating the application comprises MySQL, a relational database. JavaScript serves as the primary programming language for building this project, and REACT, a
cross-platform front-end mobile application framework, is employed to transform User Interface designs into functional interfaces. For backend interactions, the Node.js framework is used.

The backend aspect of the application involves querying multiple databases to intake and display various data received from users. Jest, a JavaScript testing framework, is employed for creating and testing all JavaScript components. Version control, issue tracking, and continuous integration are managed through GitHub. Additionally, the packaging of numerous modules and libraries is facilitated by Docker, a containerization software.

Hardware requirements for the application include an Android or Apple mobile device with internet access, an application server, and a database server. Bluetooth trackers, which help maintain the location of different items stored under a patient's account, are also required. Furthermore, motion and temperature sensors are integrated into the application for enhanced functionality.

3 Identification of Case Study

Thought Locker is a comprehensive solution designed to cater to the needs of individuals with mild to moderate dementia, as well as their caregivers or family members. The primary goal of Thought Locker is to provide a platform that enhances the lives of dementia patients and eases the burden on their caregivers.

By offering an easy method for locating lost or misplaced items, Thought Locker grants patients and caregivers a sense of relief and independence. The application also collects and retains analytics, enabling caregivers to visualize and monitor patient habits, such as the frequency of misplaced items or instances when a door has been left open.
Furthermore, Thought Locker serves as a reminder system for patients, ensuring they take their medications on time and are aware of upcoming appointments. It also provides a direct line of communication for patients to immediately contact caregivers in case of urgent needs. This cost-effective solution is an attractive alternative to hiring full-time caregivers, making it more accessible to a broader audience.

Beyond individual patients and their caregivers, Thought Locker has potential benefits for medical facilities, insurance companies, Medicaid, and Alzheimer's research groups. By leveraging the data collected and insights gained from user interactions, Thought Locker can help inform future research and improve dementia care practices.

4 Product Prototype Description

Thought Locker is designed to alleviate the concerns of families, friends, and caregivers about the well-being of individuals with dementia. By fostering a sense of independence for the dementia patient, the platform aims to reduce stress for all involved. While the prototype of Thought Locker may not include all features and may have limited versions of certain functions found in the complete product, it effectively demonstrates its potential to solve targeted challenges. This is accomplished by providing an overview of its core features and simulating the capabilities planned for the full version.

4.1 Prototype Architecture (Hardware/Software)

The prototype is designed to incorporate the majority, but not all, of the technologies selected for the final project. It's aimed at facilitating user interaction via mobile phones or tablets powered by iOS and Android, while the development
process will primarily be conducted on desktops or laptops running Linux, MacOS, or Windows. In terms of software, the prototype will be developed using languages such as JavaScript, HTML, and CSS and will leverage a MySQL database. The back-end server will be locally hosted. For testing purposes, Jest will be employed, and various APIs like React Native, Google Maps Geolocation API, Express.js, and Node.js will be integrated to enhance functionality and user experience. Additionally, to maintain streamlined development and deployment processes, tools and platforms like Git, Github, and Gitlab will be used for CI/CD and version control.

Table 1

*Thought Locker RWP vs. Prototype Features*

<table>
<thead>
<tr>
<th></th>
<th>Patient</th>
<th>Caregiver</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACCOUNT MANAGEMENT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Creation</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Account Deletion Verification</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Login / Authentication</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>User Profile Management</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>User Location Information</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Location data will be simulated</td>
</tr>
<tr>
<td><strong>USER INTERFACE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Task Completion</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Task Deletion Verification</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Location data will be simulated</td>
</tr>
<tr>
<td>Item Locator</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Item Registration</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Item Deletion Verification</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Event Viewer</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Calendar</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Sensor Monitoring</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Sensor data will be simulated</td>
</tr>
<tr>
<td>Contact Center</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Contact data will be simulated</td>
</tr>
<tr>
<td>Emergency Contacts</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Contact data will be simulated</td>
</tr>
<tr>
<td>Notifications</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Remote Access</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Not implemented in the prototype</td>
</tr>
<tr>
<td><strong>DATA MANAGEMENT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Analytics</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Analytics will be based off simulated data</td>
</tr>
<tr>
<td>Reports</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Item Tagging</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Partial. Geotag data will be simulated</td>
</tr>
<tr>
<td>Images</td>
<td>✔️</td>
<td>✔️</td>
<td>Full</td>
<td>Full</td>
</tr>
</tbody>
</table>
4.2 Prototype Features and Capabilities

In the development of the prototype, various aspects will experience different levels of implementation—some will be fully fleshed out, others will see partial development, and a few might be omitted altogether. A full implementation is anticipated for user account management, albeit with simulated location information. This includes the capability for account creation and deletion, profile and preference management, and login features.
Similarly, certain user interface features will be fully brought to life. The prototype will facilitate task management, allowing the addition, deletion, and scheduling of tasks. It will feature a calendar to accommodate appointments and tasks meant for the patient to fulfill throughout the day. The prototype will also allow the addition and deletion of items that the patient often loses and will support in-app notifications.

However, the implementation will be partial for features necessitating the use of external data sources, as the prototype aims primarily to demonstrate in-app functionalities and not necessarily to integrate real-world data. For instance, the item locator will operate on simulated item location data, generating random item locations on a map to exhibit the feature. Similarly, the contact data will be simulated, with implemented parameters for adding and deleting contacts. The process of calling and notifying an emergency contact will see full implementation, although it will employ simulated contact information.

Remote access will not be a part of this prototype as it does not align directly with addressing the challenges encountered by individuals with dementia; it is perceived more as a supplementary feature to be potentially added in future versions. Regarding data management, there will be a mix of fully and partially implemented features. User reports and item tag data will be simulated to facilitate further testing with the target market, ensuring more accurate report generation. Formatted reports and images used to track items will be completely developed, serving as a demonstrative
tool to illustrate how caregivers can gain insights from patient activity to refine their care plans accordingly.

4.3 Prototype Development Challenges

In addition to the overarching challenge posed by a constrained timeframe of a semester and a limited team size, which may potentially curtail the prototype’s functionality, the team’s general lack of familiarity with certain development tools may pose a risk of impeding the timely attainment of sprint goals. Specifically, the nuanced syntax of React Native and the commands of GitLab may present substantial learning curves, given the extensive reliance on these technologies throughout the prototype’s development. Furthermore, the endeavor to integrate front and back-end structures employing disparate development tools may elicit compatibility issues, exemplified by the integration challenges between Google Calendar and Google Maps API data with their corresponding features and the overall application.

Such integration endeavors underscore the complexities inherent in ensuring seamless interoperability amongst numerous individual components, such as those involved in the item-finding feature, where aligning Google Maps API with front-end and back-end code is pivotal. The consequential intricacies may occasionally render the achievement of certain functionalities challenging. Additionally, any discrepancies or unfinished tasks from preceding sprints can significantly stall the subsequent development phases. For instance, any misconfigurations in the prototype database during the initial sprint may potentially halt the progress of the entire back-end development team until resolutions are found.
Ensuring the prototype is architected to allow scalability beyond the use of simulated data also stands as a meticulous and crucial challenge. Additionally, selecting the most suitable tool for each specific task will be essential, especially given the general unfamiliarity with the capabilities and constraints of available development tools. This unfamiliarity may necessitate shifts in tool selection to effectively realize the product’s envisioned outcomes.
5 Glossary

**Amazon Web Services (AWS):** A cloud computing platform that provides a variety of services including compute, storage, databases, analytics, machine learning, networking, mobile, developer tools, security, and enterprise applications.

**Android:** An open-source mobile operating system based on the Linux kernel and developed by Google.

**Application Programming Interface (API):** A set of protocols, routines, and tools for building software applications that specify how software components should interact with each other.

**Application Server:** A software framework that provides an environment for running applications.

**Authentication:** The process of verifying the identity of a user.

**Biometric:** unique physical or behavioral characteristics of an individual to identify them.

**Bluetooth Low Energy (BLE):** a wireless communication technology used for short-range communication between devices.

**Caregiver/Family Member:** Any person that is able to provide assistance with managing a dementia patient’s symptoms. Their duties consist of transporting patients to and from their commitments, maintaining a patient’s daily routine, and reminding them to take their medications at regular intervals.
**Continuous Integration (CI):** A software development practice that involves frequently integrating code changes from multiple developers into a shared repository, verifying that the changes do not break the build and that the software continues to function correctly.

**Containerization:** A method of packaging and deploying software applications with all their dependencies into a single unit, called a container, which can run reliably and consistently across different computing environments.

**Database Server:** A computer program or software application that provides database services to other computer programs or clients.

**Docker:** A software platform that allows its users to build, test, and deploy applications in standardized executables resembling containers.

**GeoTag:** A physical tag that emits a Bluetooth signal to assist devices in determining their location.

**GitHub:** An open-source repository service that allows its users to work on a single project simultaneously.

**Google Maps Geolocation API:** A service provided by Google that allows developers to determine the location of a device using Wi-Fi or mobile network signals.

**iOS:** A mobile operating system developed by Apple Inc based on the Unix operating system.

**Issue Tracking:** The process of managing and resolving software issues, bugs, and feature requests.
**JavaScript**: Scripting programming language that creates dynamic web page content and mobile applications.

**JavaScript Object Notation (JSON)**: A lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate.

**Jest**: Test-runner for JavaScript applications that supports a JavaScript library for creating, running, and structuring tests.

**MongoDB**: Non-relational document database that provides support for non-relational querying.

**Node.js**: An open-source, cross-platform JavaScript runtime environment that enables the execution of server-side JavaScript code.

**Non-relational Database**: A type of database that doesn't rely on the traditional structure of tables, columns, and rows found in relational databases. Instead, they are designed to handle large and complex sets of unstructured, semi-structured, or structured data.

**Patient**: Any individual suffering from mild to moderate dementia. Their symptoms typically comprise of occasional disorientation, difficulties with making decisions, and short-term memory loss.

**PostgreSQL**: Free and open-source database management system that supports relational (SQL) and non-relational (JSON) querying.

**Programming Language**: A formal language used to communicate instructions to a computer or other machine.
**React**: An open-source JavaScript library that is used to build user interfaces for web and mobile applications.

**Relational Database**: A type of database that stores data in a structured format, using rows and columns to represent data entities and attributes.

**Relational Database Management System (RDBMS)**: A software system that enables users to create, maintain, and manipulate relational databases.

**Repository**: A central location where digital files, usually in the form of software source code, are stored and managed.

**Structured Query Language (SQL)**: A programming language used to manage and manipulate data in a relational database management system (RDBMS).

**Testing Framework**: A set of guidelines, standards, and tools that software developers use to create and run automated tests for their code.

**Version Control**: A system that tracks and manages changes to software code, documentation, and other files over time.
6 References


mph, s. c., md. (2020, february 14). what’s the best way to manage agitation related to

harvard.edu/blog/whats-the-best-way-to-manage-agitation-related-to-dementia-20200214

18816

recallcue connected day clock for seniors with dementia. (n.d.). recallcue. retrieved april

3, 2023, from https://www.recallcue.com/