Lab 1 – Thought Locker Product Description

Ryan Wilkinson
Old Dominion University
CS 411W, Fall 2023
Professor J. Brunelle
September 15, 2023
Version 2 Rough Draft
Table of Contents

1 Introduction...........................................................................................................................................3
2 Thought Locker Product Description.....................................................................................................5
   2.1 Key Product Features and Capabilities............................................................................................5
   2.2 Major Components (Hardware/Software)..........................................................................................7
3 Identification of Case Study......................................................................................................................8
4 Thought Locker Product Prototype Description....................................................................................10
   4.1 Prototype Architecture (Hardware/Software)...................................................................................10
   4.2 Prototype Features and Capabilities...............................................................................................11
   4.3 Prototype Development Challenges...............................................................................................13
5 Glossary..................................................................................................................................................15
6 References...............................................................................................................................................19

List of Figures

1 Major Functional Component Diagram..................................................................................................8
2 Prototype Major Functional Components Diagram................................................................................10

List of Tables

1 Thought Locker RWP v. Prototype Features.........................................................................................12
1 Introduction

Dementia is an umbrella term used to describe individuals that have an impairment in their ability to remember, think, and make decisions. It encompasses a wide variety of neurological conditions and diseases, including Alzheimer’s disease, vascular dementia, and Lewy Body. This cognitive impairment is common among older populations of adults, affecting one in ten Americans aged 70 and older. The proportion of adults that are diagnosed with a variation of dementia increases significantly if categorized by age. While only 3% of adults aged 70-74 have dementia, this percentage rapidly increases to 33% of adults aged 90+ (PRB, n.d.).

Despite the large number of adults that are diagnosed with the condition, not all cases are equivalent in terms of the severity of symptoms. According to research performed by the National Institute of Health, roughly 50% of Alzheimer’s patients had a mild form of the disease, while 30% and 19% received moderate and severe diagnoses, respectively (National Institute of Health, 2021). Mild dementia is characterized by several distinct symptoms, such as short-term memory loss, difficulties with communicating and making decisions, challenges with spatial reasoning and visual processing, and changes in mood and temperament. During this phase of dementia progression, the independence of the individual is largely preserved, as these symptoms are not severe enough to require constant caregiver supervision (Alzheimer’s Society, n.d.).

The predictable result of developing dementia is the restrictions it imposes on daily life. Short-term memory loss can cause individuals to become more prone to losing household items, such as keys and reading glasses. These events can worsen mood swings and inadvertently create schisms between the individual and their loved ones.
Dementia can also impair an individual’s ability to plan and coordinate their activities. Often, patients must remember many important commitments, such as doctor’s appointments, reminders to take life-saving medication, holidays, and milestones achieved by family members. Because dementia can cause orientation to be impeded, these activities are often misremembered and sometimes forgotten entirely. This can lead the patient to become distressed or even sick if doctor’s instructions are not heeded correctly.

The progression of dementia can also cause caregivers to feel helpless in caring for their patients. Alongside their own responsibilities, caregivers must maintain their patient’s schedule, remind them of upcoming appointments, and ensure that the patient is completing their daily obligations. This level of responsibility can often be taxing for family members that are not hired full-time caregivers. Caregivers in many cases also lack the ability to monitor their patient’s activity in real-time. This can lead to the possibility of catastrophic consequences, such as the patient leaving their home unattended or leaving a door unlocked or opened.

Due to the short-term memory loss and lack of orientation that is often associated with dementia, patients can also forget how to contact loved ones in the event of an emergency. A typical touchscreen interface on a smartphone or smart watch can be complex to those who don’t use these technologies often. This initial confusion can lead to frustration, as patients may resort to repeatedly call the wrong number, inadvertently misdial contacts, or accidentally call emergency services.

These issues, when viewed together, point towards the need for a centralized platform that can strike a balance between patient independence and caregiver monitoring capabilities. A proposed solution would allow for easy-to-use tracking of the locations of important items, provide
caredgers with a comprehensive tool to monitor patient activity, and allow patients to maintain a
degree of independence while still being cared for properly.

Thought Locker is an application designed to strengthen and enhance the patient-caregiver
relationship. It streamlines the item-finding process, allows caregivers to effectively monitor
patient activity, and enables patients to retain a degree of independence despite the mental
challenges they face each day.

2 Thought Locker Product Description

Thought Locker is a mobile application designed to assist patients with mild to moderate
dementia and their caregivers. The purpose of the application is to simplify finding solutions to
the day-to-day challenges and disruptions that individuals living with mild to moderate dementia
and their caregivers may experience. It is designed with two primary intentions in mind: to
maximize the ability of the caregiver to monitor a patient’s activities and to grant a degree of
personal freedom for patients who still wish to remain independent of a full-time caregiver.

2.1 Key Product Features and Capabilities

Thought Locker has several key features that simplify the patient-caregiver relationship.
These features can be customized based on the patient’s needs. The first is a simple calendar and
reminder interface. The caregiver is primarily in-control of this aspect of the application, but the
patient can also add and delete events as needed. This feature functions in much the same way as
a normal calendar and reminder application, where the caregiver or patient can customize the tasks
that need to be fulfilled. As those tasks are checked off and completed, the patient and caregiver
will receive a notification through their device. This feature supports patients with short-term
memory loss and sends them a friendly reminder if a task is not completed within the preset time
they specified. The structure of this feature ensures that patients can complete tasks independently while under the watchful eye of a caregiver.

Another feature Thought Locker houses is its contact center. Rather than a large rolodex-like contact system in a typical iOS device, Thought Locker instead creates a simple interface with a select-few contacts that are important to the patient. A caregiver or patient can add emergency contacts as needed. In the event a patient is in need, they can simply press a button that is always present within the application that calls their highest-priority emergency contact. In the event they do not answer to the notification, the next-highest individual on the priority list is contacted. This feature will be created to tackle the orientation and short-term memory difficulties that individuals with dementia experience. Rather than fumbling through various applications, patients can instead readily contact who they need to without having to navigate the complex array of applications on their device. This simplification inherently eliminates much of the frustration those with dementia experience.

One of the features that truly separates Thought Locker from other competitors is its item location feature. Caregivers can pair important items with Geo Tags to the patient’s device, such as their car keys or medicine bottles, so that the patient or caregiver can readily find them when they are needed. Rather than spending hours trying to find important medication, the caregiver or patient can instead use the application to “ping” the item’s last location and have it play a sound. Caregivers will also be able to log the locations of important items so that in the event they cannot find an item, they can try looking in specific locations that are often visited by the patient. This feature will allow patients and caregivers to find items more readily, reducing the potential for conflicts related to misplaced items.
Finally, Thought Locker will enable caregivers to effectively respond to patient needs using comprehensive monitoring features. A common complication associated with the loss of orientation in dementia patients is forgetting to complete important actions, such as turning off a stove, locking a door, or closing a refrigerator door. Thought Locker enables the caregiver to know when these events happen without having to be present with the patient. Using sensors on doors and appliances, Thought Locker can track when doors are not locked or when appliances are not turned off. The application will produce analytics based on these activities, which the caregiver can use to tailor the patient’s environment to mitigate risk.

These features, incorporated into one application, can mitigate the risks on behalf of the patient and caregiver and can allow the patient to retain a degree of independence while still being properly cared for.

2.2 Major Components (Hardware/Software)

Thought Locker is an application intended to be cross-compatible across multiple devices, such as smartphones, tables, smart watches, or any other devices connected to Android and iOS operating systems. An application and database server will also be needed to house the analytics. To sign up, a patient simply creates an account with their email address and password. A caregiver must create a separate account with their own email and password. When patients first interact with the software, they will be presented with the option to add their caregiver. Once the caregiver is added, they will have administrative control of what the patient is able to do within the application.

The major components of the application according to these user roles are shown in Figure 1. As stated before, the caregiver will have primary responsibility for creating calendar reminders,
creating the emergency contact list, pairing items for the patient to then interact with, and monitoring their daily habits.

**Figure 1**

*Major Functional Components Diagram*

Thought Locker also incorporate several types of software to achieve its goals: AWS for backend server development, PostgreSQL and MongoDB as databases, and JavaScript as the primary programming language. Jest will be used for testing, Docker will be used for containerization, and GitHub will be used for version control, issue tracking, and CI/CD.

### 3 Identification of Case Study

The target demographic of this product are individuals that have mild to moderate dementia and their respective caregivers and family members. There is currently no popular application on the market that is designed specifically for caregivers to effectively manage patient independence, so this product would be intended to support caregivers and patients in managing dementia.
symptoms. Patients will be able to plan tasks ahead of time, call their contacts at any time, or find important items themselves. Caregivers, by contrast, will be able to leave more tasks up to the discretion of the patient rather than claiming comprehensive responsibility of the patient’s everyday activities.

In terms of a case study group, the product would seek feedback from individuals with mild to moderate dementia, along with the caregivers who support them. They will provide valuable feedback for what daily habits should be monitored by caregivers, as well as the capabilities of patients in this phase of the dementia progression. This case study would also hone Thought Locker’s analytic capabilities, as it would tailor the product to the most popular needs of caregivers and their patients. Their feedback could also be used to offer insights on the design of the user interface, the quality of the design features, and making sure that the product truly meets the needs of its target market.

While Thought Locker is specifically designed for dementia patients and their family members/caregivers, the product could potentially be used by other population segments, such as those with general short-term memory problems or other individuals with other conditions that may require monitoring by a trusted party. By design of the application, a patient does not necessarily need a caregiver as an administrator. The caregiver is offered to take this role in the initial set-up of the application. Otherwise, the patient, by default, has all of the capabilities of a caregiver. In terms of monitoring capabilities, Thought Locker may allow those with mental challenges the opportunity to live by themselves under the auspices of a trusted third-party.
4 Thought Locker Product Prototype Description

Thought Locker’s prototype design is intended to provide patients and caregivers with an effective tool to alleviate common issues that often strain the patient-caregiver relationship and ultimately mitigate stress for both parties. Although Thought Locker’s prototype will not contain some features and others will be limited in their functionality, its prototype will demonstrate key features that can effectively accomplish the main objectives of the overall project.

4.1 Prototype Architecture (Hardware/Software)

A pictorial representation of the major components of Thought Locker’s prototype is shown in Figure 2. The structure of the prototype MFCD remains largely the same when compared to the general MFCD, with the patient and caregiver interfaces remaining separate. However, there are some key differences in how the features of Thought Locker will be presented in the final demonstration.

Figure 2

Prototype Major Functional Components Diagram
Thought Locker’s prototype will make key adjustments in its hardware and software components. From a hardware standpoint, the prototype will be designed for Android use only rather than both Android and iOS due to the team’s strength with the React programming language. Thought Locker will also employ MySQL as its database server rather than PostgreSQL or MongoDB due to its familiarity among the development team and limited financial cost. All other elements of the prototype, such as the languages used, back-end server, testing framework, API, and CI/CD will mirror the real-world product.

4.2 Prototype Features and Capabilities

Many of Thought Locker’s functionalities represented in the prototype will remain consistent with the real-world product. As seen in Table 1, much of the product’s account management, user interface, and data management features will resemble the real-world application. Most of these features are crucial to the development of a usable application for the application’s target market, such as the ability to create and manage an account, interact with event planning and completion features, and the ability to add and delete items to be found later. Without a full implementation of these features, both caregivers and patients would likely experience frustration, as the prototype would feel less like a usable application and more like an accumulation of tangentially relevant ideas.
It should be noted that some functional elements will be partially implemented with simulated data. Due to the nature of the application sharing large amounts of identifiable information, such as personal contacts, many potentially relevant datasets are often walled-off from use by potential developers, citing the highly confidential information contained within them. A similar case can be made for the case of patient and item location information, as it too can be considered private. Other features, such as sensor monitoring and geotag information, will also use simulated data to limit time spent sifting through the large amount of information collected by these sensors. Because user analytic reports are largely reliant on data collected by these sensors, it too will be simulated in the Thought Locker prototype.

In terms of risk mitigation, the prototype will implement many of the same strategies as the real-world product. If the patient loses engagement in Thought Locker or is unable to use its functions, reminders and hints will be provided to maintain more consistent interaction with the

Table 1

**Thought Locker RWP vs Prototype Features**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Patient</th>
<th>Caregiver</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNT MANAGEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Creation</td>
<td>✓</td>
<td>✓</td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Account Deletion Verification</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Login / Authentication</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>User Profile Management</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>User Location Information</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>USER INTERFACE</td>
<td></td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Task Scheduling</td>
<td>✓</td>
<td>✓</td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Task Completion</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Task Deletion Verification</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Item Locator</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Item Registration</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Item Deletion Verification</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Event Viewer</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Full: Location data will be simulated</td>
</tr>
<tr>
<td>Calendar</td>
<td>✓</td>
<td>✓</td>
<td>Full</td>
<td>Full: Location data will be simulated</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>Partial: Contact data will be simulated</td>
</tr>
<tr>
<td>Remote Access</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Not implemented in the prototype</td>
</tr>
<tr>
<td>DATA MANAGEMENT</td>
<td></td>
<td></td>
<td>Full</td>
<td>Partial: Analytics will be based on simulated data</td>
</tr>
<tr>
<td>User Analytics</td>
<td>✓</td>
<td></td>
<td>Full</td>
<td>Partial: Geotag data will be simulated</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>
application. In the event of an information breach or HIPAA violation, a license agreement and encryption algorithm will be designed to enhance user security. Technical risks associated with the lost item feature, such as the inability to authenticate a found item or locate a lost item, will be addressed by tapping the geolocation device rather than image confirmation and sending a lost item report to the caregiver. In the event that patient connection loses connection to their calendar temporarily, a cache of the information contained within it will be locally stored on their device.

If the principal objectives set in Table 1 are met, the prototype can be considered successful despite the extensive use of simulated data because one of the functional goals of the product in this phase is to showcase its potential in the space of finding lost items and aiding patients in their daily lives. The developers also intend to create a useable application that can be tested and refined to meet the needs of the target market more properly. The prototype will be limited in its use of real-time data and how it interfaces with external technologies, but the features of the product will be shown to be inherently beneficial. In meeting these objectives, a degree of patient independence will be achieved while also maintaining the crucial supportive role of caregivers.

4.3 Prototype Development Challenges

Besides the more general challenge of the relatively short time constraint of one semester and a smaller team size that may limit the level of functionality of the prototype, a general unfamiliarity among the team of some prototype development tools may lead to delayed achievement of sprint goals due to the learning curve involved. Compatibility issues may also arise when attempting to tie together certain front and back-end structures that utilize different development tools. There may also be times when it may be difficult to achieve a desired functionality due to the large number of individual components that must work together, such as the item-finding feature.
Other challenges that may be presented include being unable to initiate the next sprint without the completion of the prior sprint; for example, if the prototype database was not configured correctly in the initial sprint, the entire back-end development team may be stalled until the issue can be resolved. Ensuring that the prototype is developed in such a way that it can be scaled beyond the simulated data it is fed will also pose a tedious but necessary challenge. Finally, ensuring that the right tool is being used for the most relevant task will also be a challenge, as a general unfamiliarity with the tool’s capabilities and limitations may lead to a selection of another development tool to effectively achieve the product’s desired result.
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


