Lab 1 – Thought Locker Product Description

Ty Schiffer
Old Dominion University
CS411
Professor Brunelle
09/29/2023
Version 3
LAB 1 – Thought Locker Product Description

Table of Contents

1 Introduction .................................................................................................................................................. 2

2 Product Description ..................................................................................................................................... 4
   2.1 Key Product Features and Capabilities ................................................................................................. 5
   2.2 Major Components (Hardware/Software) .............................................................................................. 6

3 Identification of Case Study ....................................................................................................................... 8

4 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 ..................................................................................................................................................... 14

6 References .................................................................................................................................................. 18

List of Figures

Figure 1: Thought Locker Major Functional Component Diagram ................................................................. 7

Figure 2: Thought Locker Major Functional Component Implementation Diagram ..................................... 11

Figure 3: Thought Locker Feature Chart ..................................................................................................... 12
1 Introduction

According to Alzheimer's Disease International (2020), one in ten Americans aged 70 and older have some form of dementia, and the number of individuals diagnosed with dementia is steadily increasing over the years. Furthermore, of the dementia patients diagnosed, 50.4% of cases are mild, 30.3% are moderate, and 19.3% are severe. Individuals with mild dementia diagnosis may need very little assistance in daily living activities and may experience changes in mood, memory problems, and difficulty effectively planning and thinking things through (Alzheimer’s Society, n.d.). On the other hand, individuals with moderate dementia may need frequent reminders and some assistance with washing and dressing. Moreover, they may develop paranoia, experience intense symptoms of anxiety and depression, and memory problems may worsen (Alzheimer’s Society, n.d.).

In the case of mild or moderate dementia, individuals would prefer to maintain some of their independence (Alzheimer’s Society, n.d.). However, as dementia symptoms worsen, assistance is required from either a family member or an outside source, and care becomes stressful for the caregiver/family member due to the required constant attention (MPH, 2020). It is, therefore, crucial to explore innovative ways to support dementia patients and their caregivers to enhance their quality of life.

As the number of individuals diagnosed with dementia continues to increase, there is a need for a solution that can help maintain their independence while also providing necessary support. That is where Thought Locker comes in - a mobile assistant designed to help dementia patients maintain their independence through reminders, item location, monitoring, and analytics that will be provided to the caregiver.
The solution to the societal problem of dementia must address the needs of patients as they progress through different stages of the disease, providing support and assistance as required. It must also consider the needs of caregivers and family members, who can become overwhelmed by the constant demands of caring for a loved one with dementia. Thought Locker provides a comprehensive solution that can address these needs, allowing patients to maintain their independence while also providing the necessary support and assistance for caregivers.

The growing problem of dementia requires innovative solutions that can address the needs of patients and caregivers alike. Thought Locker also addresses this need by providing a comprehensive mobile assistant that can help patients maintain their independence while also providing the necessary support and assistance for caregivers.

2 Product Description

Thought Locker is an innovative application designed to assist individuals living with dementia. It provides a range of tools and features that help users stay organized and retain their independence, even as their symptoms worsen over time. The application includes a comprehensive suite of tools, such as an item finder, task manager, event viewer, and calendar, that allow users to manage their daily responsibilities and stay on top of their schedules. With its user-friendly interface and intuitive features, the application helps users with dementia stay organized and on track, reducing stress and improving their overall quality of life. Thought Locker is an application designed to assist both individuals living with dementia and their caregivers in managing the challenges associated with the condition. It aims to help patients maintain their independence by providing tools to keep them organized and focused on their daily tasks.
2.1 Key Product Features and Capabilities

Thought Locker is a comprehensive dementia application that is designed to improve the communication and organization of tasks between caregivers and patients. The application’s key features include an Item Finder, a Calendar, an Event Viewer, and a Contact Center.

The Item Finder feature of Thought Locker allows patients to locate items using a Bluetooth tracker API. This feature is particularly useful for individuals who have memory issues and tend to misplace their belongings. With this feature, patients can easily locate their belongings, such as keys, wallets, or phones, with the help of a tracker that is connected to the application.

The Calendar feature of Thought Locker utilizes Google's API to allow caregivers to keep track of appointments, meetings, and other important events. This feature enables caregivers to manage their schedules effectively and avoid any potential conflicts. The application also allows caregivers to set reminders for events, ensuring that they do not forget any important appointments.

The Event Viewer feature of Thought Locker enables caregivers to monitor and track tasks completed by other caregivers, as well as the patient's condition. The application allows caregivers to set alerts for low, medium, or high priority tasks, ensuring that important tasks are not overlooked. This feature also helps caregivers to identify any potential issues early on, allowing them to take necessary actions promptly.

The Contact Center feature of Thought Locker enables patients to contact emergency services or family members using a priority queue system. This feature is particularly useful in emergency situations when patients need to contact their loved ones or seek medical attention.
urgently. The application's priority queue system ensures that important messages are given priority, allowing patients to receive the help they need in a timely manner.

The patient's role within the application is to use the features provided by the application to improve their quality of life and manage their daily tasks. However, the patient's access to certain features may be limited or delegated to other caregivers if their cognitive function or memory declines. This ensures that the patient is not overwhelmed by the application's features and can continue to use it effectively.

In contrast, the caregiver's role within the application is to manage the patient's account and ensure that the application is tailored to the patient's needs. The caregiver has the ability to control the patient's access to features, delegate tasks to other caregivers, and monitor the patient's condition through the application. This allows the caregiver to effectively manage the patient's care and ensure that their needs are met.

By defining these roles and permissions within the application, Thought Locker creates a clear and organized system for managing caregiving tasks. The application's features and customizable settings provide caregivers and patients with a personalized and effective tool for managing daily tasks and improving quality of life.

2.2 Major Components (Hardware/Software)

Thought Locker is a mobile application that can be used on both Android and Apple mobile devices. It requires internet access to function properly. The major components of Thought Locker, as illustrated in Figure 1 are a mobile device that utilizes an application database, several APIs, and an analytic database to store activity information from the application’s vast feature set.
Figure 1:

Locating items is accomplished using the React-Native Bluetooth API that pairs with tags attached to important items. Haptic notifications are used to indicate closing or opening proximity.

Calendar notifications and reminders will utilize Google Calendar API to provide the patient a way to view upcoming events.

The application's hardware requirements include an Android or Apple mobile device with internet access, an application server, and a database server. The software requirements for the backend server are Amazon Web Services (AWS) and PostgreSQL or MongoDB databases. The programming language used for development is Javascript with the React/Node.js framework. Jest is used for testing, GitHub for repository and version control, and GitLab for continuous integration/continuous delivery (CI/CD). Docker is used for containerization.
3 Identification of Case Study

Objective: The primary objective of this case study is to demonstrate Thought Locker's features and capabilities by introducing a small sample group of users, consisting of individuals with mild to moderate dementia and their caregivers, who will serve as a beta test group. This group will help illustrate how Thought Locker can effectively assist dementia patients and their caregivers in managing daily tasks and improving overall quality of life.

Example Group: The sample group in this case study will include three individuals with mild dementia, three individuals with moderate dementia, and three caregivers who have various relationships with the patients, such as family members or professional caregivers. This diverse group will help showcase the adaptability of Thought Locker to different types and stages of dementia, as well as the varying needs of caregivers.

Purpose: The case study serves as a transition from the Real-World Product description contained in Sections 1 and 2 to the Prototype Description in Section 4. The main points to convey in this case study include:

The case study addresses dementia patients and their caregivers, who play the roles of users and customers. The case study aims to demonstrate Thought Locker's key features and capabilities, such as the Item Finder, Calendar, Event Viewer, and Contact Center, and how these features benefit the participants.

The "use case" scenario describes how the prototype will demonstrate the application's features, such as item finding, scheduling, and contacting, for individuals with mild to moderate dementia and their caregivers.
A "full capability" implementation may include the ability to integrate with other dementia care resources, services, or devices, and expand its user base to medical facilities, insurance companies or Medicaid, and Alzheimer's research groups.

The broader user and customer base may include other dementia care providers, healthcare professionals, and organizations that support dementia patients and their caregivers.

Case Study Scenario: The case study will follow the daily life of the sample group as they use Thought Locker to manage their daily tasks, stay organized, and maintain their independence. The prototype will demonstrate how patients can locate misplaced items using the Bluetooth tracker, how caregivers can monitor patients' progress and set reminders for important events, and how patients can easily contact their caregivers when needed. The case study will also highlight how Thought Locker can help caregivers analyze patients' habits and patterns to provide better care and support.

[ This space intentionally left blank. ]
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 Thought Locker prototype showcases a well-defined architecture segmented into multiple modules. This application is designed for both Android and iOS platforms, with no current development for desktop interfaces. It employs a combination of analytic tools and cloud database for storing pertinent data. As illustrated in Figure 2, while certain aspects of the software are simulated, their functionalities remain intact. Item Finder leverages simulated Bluetooth tracking, aiding users in locating misplaced items. Meanwhile, the Calendar module integrates with the Google Maps Geolocation API for enhanced event planning and management. Additionally, the Event Viewer is designed for users to track tasks and observe patient conditions, and the Contact Center allows simulated interactions, mimicking calls to family or emergency services.

User interaction is streamlined through an intuitive UI that bridges the user with these diverse modules. Supporting this front-end is a simplified backend infrastructure, ensuring smooth data flow and enhancing the overall user experience. To further clarify the ties between these elements, the prototype will come with a Major Functional Component Diagram (MFCD). This visual aid will map out the relationships between the UI, simulated Bluetooth tracking, Calendar API, Notification System, and backend operations.
4.2 Prototype Features and Capabilities

Thought Locker is an application solution devised to confront the challenges faced by dementia patients and their caregivers. Drawing from alarming statistics on the growing dementia patient population, it becomes evident that innovative solutions are essential. Thought Locker rises to this challenge, offering features to bolster patient autonomy and ensure caregiver efficiency. The application addresses common problems, such as misplacing items, through its Item Finder that utilizes Bluetooth technology. Moreover, it provides an organized system for caregivers via the Event Viewer to set and oversee tasks. In terms of scheduling and timely reminders, the Calendar taps into Google's API, ensuring patients and caregivers remain informed about appointments and crucial events. If patients need immediate attention or assistance, the Contact Center allows for swift communication. When we consider the broader societal challenge of dementia, Thought Locker serves as a comprehensive tool, aiming to enhance the daily lives of patients and provide invaluable support for their caregivers. The Thought Locker Feature Table below shows the features that are planned for the prototype.
Figure 3:

<table>
<thead>
<tr>
<th>ACCOUNT MANAGEMENT</th>
<th>Patient</th>
<th>Caregiver</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER INTERFACE</th>
<th>Patient</th>
<th>Caregiver</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<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>Full</td>
</tr>
<tr>
<td>Item Locator</td>
<td>✓</td>
<td>✓</td>
<td>Full</td>
<td>Partial: Location data will be simulated</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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA MANAGEMENT</th>
<th>Patient</th>
<th>Caregiver</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<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>

As shown in the figure above, certain features of this application will be emulated to demonstrate functionality. For example, user data, such as contacts for each role will be fabricated. Instead of using Geotagging, item tagging will be simulated by sending a ping for an item and receiving a response. Depending on where the item is simulated to be will determine the length of the notification sound to determine the location of the object. User location information will have to be emulated because there will not be real time location available for individual users.

[ This space intentionally left blank.]
4.3 Prototype Development Challenges

Constructing the prototype comes with its own set of hurdles. The team's lack of experience with React Native and Node.js will make our learning process more demanding, which might lead to unforeseen delays. During the development phase, we could face complications beyond our current expertise, necessitating either modifications or seeking external advice. Given that half of our team is employed full-time, we are restricted in terms of time, impacting the hours we can invest in the project. Setting up an AWS RDS accessible beyond the AWS VPC poses challenges in ensuring it's available to multiple devices due to IP address constraints. Simulating data, especially for sensor monitoring and displaying trends in the event viewer feature, will be challenging to represent accurately.

[This space intentionally left blank.]
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


*How to find out if an AirTag is tracking you.* (n.d.). ZDNET. Retrieved April 3, 2023, from https://www.zdnet.com/article/how-to-find-out-if-an-airtag-is-tracking-you/


18816