Lab 2 – Thought Locker Product Specification

Ryan Wilkinson

Old Dominion University

CS411W

Professor J. Brunelle

October 22, 2023

Section 1 & 2, Version 1
Table of Contents

1 Introduction..............................................................................................................3
  1.1 Purpose.............................................................................................................3
  1.2 Scope...............................................................................................................4
  1.3 Definitions, Acronyms, and Abbreviations......................................................6
  1.4 References.......................................................................................................10
  1.5 Overview.........................................................................................................12

2 General Description................................................................................................12
  2.1 Prototype Architecture Description.................................................................12
  2.2 Prototype Functional Description.....................................................................14
  2.3 External Interfaces............................................................................................17

List of Figures

Figure 1: Thought Locker Prototype Architecture Diagram.................................13

List of Tables

Table 1: Thought Locker Prototype Features Table.................................................14
1 Introduction

Dementia is a relatively common cognitive impairment that affects one in ten Americans aged 70 and older (PRB, n.d.). Individuals that develop dementia most frequently experience symptoms of short-term memory loss, difficulties with communication and decision-making, challenges with spatial reasoning and visual processing, and changes in mood and temperament (Alzheimer’s Society, n.d.). Due to its progressive nature, its severity can vary markedly from person to person, with a majority experiencing mild to moderate symptoms (National Institute of Health, 2021).

Despite a mild or moderate diagnosis, dementia can impose significant restrictions on daily life. Individuals may be more prone to losing household items, failing to remember appointments and activities, and misremembering contact information. Caregiver intervention is often needed to alleviate the burden. Due to the significant time commitment involved and a general lack of information needed to understand their patient’s habits, this delicate relationship can quickly become fractious. However, this strain can be alleviated in-part through the use of the Thought Locker application.

1.1 Purpose

Thought Locker is a mobile application intended to assist patients with mild to moderate dementia and their caregivers. The purpose of the product is two-fold: 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. It offers several functionalities to mitigate the conflicts commonly experienced by both parties. A simple calendar and reminder interface mitigates the risk of patients losing track of important events and tasks to complete.
Thought Locker’s emergency contact center centralizes critical contact details and reduces the need for the patient to open multiple applications on their device. The application’s item-finding feature also enables patients to find common household items rather than requiring caregiver assistance. Caregivers will also benefit from Thought Locker’s capabilities to monitor patient activity. Their ability to add and edit events, reminders, and findable items will grant them the ability to communicate with the patient while they are not directly present. From these features, caregivers can also more thoroughly understand patient habits through printable analytic reports. While Thought Locker will be used to supplement the patient-caregiver relationship, it is not designed to replace caregivers entirely. The product was also not designed to alleviate pressures experienced by patients with severe dementia.

1.2 Scope

The Thought Locker prototype is designed with the goal of alleviating common patient-caregiver relationship stressors. From the patient perspective, Thought Locker seeks to maintain their independence by enabling patients to find lost items at their leisure. The application also enables patients to complete their daily tasks without the presence of a caregiver. By contrast, caregivers will have the benefit of more information to effectively tailor their patient care plan. This information is customizable to the caregiver, whether that be through notification alerts or printable reports.

The Thought Locker prototype will demonstrate many of the same features as outlined in the real-world product. However, some features will be eliminated and others will be limited in their functionality to most efficiently deliver the overall goals and objectives of the product. For example, the device login screen and databases will be fully implemented as well as the calendar and reminder feature. Other features, such as the item-finder, emergency contact center, and the
analytics tied to these features will be populated with simulated data. Despite this substitution, the overall goal of delivering a product that will improve the quality of the patient-caregiver relationship will still be accomplished.

The prototype implementation will also conform to the data management risk mitigation strategies that were pre-established by the development team. All private data, including patient and caregiver inputs and login information, will be fully implemented and properly stored within the database with encryption and database access control. Any element that would require data collection with a source external to the application, such as Bluetooth information and item location information, will remain simulated.

[This space is intentionally left blank.]
1.3 Definitions, Acronyms, and Abbreviations

**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.

GitLab: 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.

**Jira**: A project management and issue tracking tool developed by Atlassian. It is widely used by software development and project management teams to plan, track, and manage their work.

**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 consist 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.

[ This space is intentionally left blank.]
1.4 References


[This space is intentionally left blank.]
1.5 Overview

This product specification provides the hardware and software configuration, external interfaces, product features, and test case simulation of the Thought Locker prototype. Information contained within the remaining portions of this document will include a description of each application feature and their respective implementation requirements.

2 General Description

The Thought Locker prototype will be deployed as a mobile application for iOS and Android. The prototype will include algorithms for the login, item finder, task manager, emergency contact center, event viewer, and analytic reports. Each of these features will be demonstrated to the extent that the core functionality meets the application’s intended purpose and is shown to be effective. As a result, some features will employ the use of simulated data, particularly in the contexts of item location and pairing, sensor monitoring, emergency contacts, and analytic reports.

2.1 Prototype Architecture Description

Thought Locker’s prototype shall include primarily in-house components that interface with selected external APIs. Thought Locker’s in-house elements include a patient and caregiver interface, a MySQL cloud and analytic database housed on an ODU remote server, and a mock patient-caregiver relationship with all features populated for demonstration purposes. Thought Locker will also employ React Native to construct the user interface elements, as well as the Google Maps and Calendar API to house the locations of mock items and patient events, respectively. The major components of the product are shown in Figure 1.
Thought Locker’s patient interface shall consist of a login page and dashboard. Once proper credentials are entered, the patient will be directed to a dashboard that provides them with options to view scheduled tasks and events, locate items, or access the emergency contact center. Caregivers will undergo the same login process, bringing them to their own separate dashboard. The caregiver interface will share many of the same features as the patient, including task and event management and item finding. However, the caregiver will be able to make modifications to these events and tasks by default, while the patient interface must be granted permission. The caregiver interface will also exclusively contain an analytics tab, which can be used to summarize patient activity.

The cloud and analytic databases that will be stored on an ODU server shall contain patient and caregiver input data. Most input data will be entered by the caregiver, while the completion status of tasks can also be changed by the patient. Mock input data will be used to populate each
feature within the MFCD, which will be viewable within the user interface and the back-end database.

2.2 Prototype Functional Description

The core functionalities of Thought Locker will be included in the prototype. Most features will be implemented fully, while others will be partially implemented or use simulated data. Thought Locker’s primary features consist of a task scheduler, item locator, calendar, emergency contact center, and analytic report generator for the caregiver. Table 1 outlines the features of the Thought Locker prototype and the extent they are implemented.

Table 1

Thought Locker Feature Description and Prototype Implementation

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
<th>Prototype Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACCOUNT MANAGEMENT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Creation</td>
<td>Set up a new patient or caregiver account</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td>Account Deletion Verification</td>
<td>Ensure the intentional removal of a user's account, requiring confirmation</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>steps to prevent accidental deletion</td>
<td></td>
</tr>
<tr>
<td>Login / Authentication</td>
<td>Users prove their identity and gain access to the application by providing</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>the required credentials</td>
<td></td>
</tr>
<tr>
<td>User Profile Management</td>
<td>Administration of individual user accounts within the application, allowing</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>users to customize their settings, preferences, and personal information</td>
<td></td>
</tr>
<tr>
<td>User Location Information</td>
<td>Identifies the geographic position of a user through their device by</td>
<td>Partially Implemented:</td>
</tr>
<tr>
<td></td>
<td>enabling location-based services</td>
<td>Location data will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>simulated</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Implementation Status</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Task Scheduling</td>
<td>Organize specific tasks to be executed at predetermined times</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td>Task Completion</td>
<td>Mark a task complete after successfully executing it</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td>Task Deletion</td>
<td>Ensure the intentional removal of a specific task, requiring confirmation</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>steps to prevent accidental deletion</td>
<td></td>
</tr>
<tr>
<td>Item Locator</td>
<td>Identifies the location of items, making it easier to find and manage them</td>
<td>Partially Implemented: Location data will be simulated</td>
</tr>
<tr>
<td>Item Registration</td>
<td>Record and document information about an item in a database, including</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>details such as ownership, specifications, and relevant data for tracking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and management</td>
<td></td>
</tr>
<tr>
<td>Item Deletion</td>
<td>Ensure the intentional removal of a specific item from the database,</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>requiring confirmation steps to prevent accidental deletion</td>
<td></td>
</tr>
<tr>
<td>Event Viewer</td>
<td>Provides logs and details about system, security, and application events,</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>aiding in system troubleshooting, monitoring, and maintenance</td>
<td></td>
</tr>
<tr>
<td>Calendar</td>
<td>Visual tool that organizes and displays dates, days, and events, facilitating</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>time management and scheduling</td>
<td></td>
</tr>
<tr>
<td>Sensor Monitoring</td>
<td>Continuous observation and collection of data from sensors to track and</td>
<td>Partially Implemented:</td>
</tr>
<tr>
<td></td>
<td>assess changes to item movement in the environment</td>
<td>Sensor data will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>simulated</td>
</tr>
<tr>
<td>Emergency Contacts</td>
<td>Individuals designated to be contacted in urgent or critical situations,</td>
<td>Partially Implemented:</td>
</tr>
<tr>
<td></td>
<td>typically for medical, safety, or personal reasons</td>
<td>Contact data will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>simulated</td>
</tr>
</tbody>
</table>
Most account management features, such as account creation and deletion, the login process, and profile settings will be fully implemented within the prototype. Patients and caregivers will input their account data into the login screen, which will be cross-referenced with the MySQL database for verification. New account data will be stored into the database as needed. User location information, which is intended for use by the caregiver to find a patient, will utilize simulated Google Maps API data, as collected data is not entirely necessary from a prototyping perspective.

The task manager and calendar user interface components will be fully implemented to be used by caregivers to remind patients of upcoming events and tasks to complete. Aspects of other features, such as the item finder, will fully implement components that do not require simulated data, such as registration, deletion, and image upload. Item location information, which would be
used by the patient or caregiver to find a lost item, will use simulated data, as exact locations of items do not need to be showcased to demonstrate that the item finding process operates effectively. The emergency contact feature, which will be used by patients to call their most relevant contacts when needed, will utilize simulated contact information due to privacy restrictions concerning utilizing actual phone numbers and names of individuals. The fully implemented event viewer and notification infrastructure, which summarizes activity across the application, will be used by the patient and caregiver to pinpoint behavioral patterns that may contribute to sources of stress. Simulated sensor data will be included in the prototype to showcase the potential for Thought Locker to expand its analytic capabilities beyond the calendar and reminder interface.

Analytics, such as task completion percentage, triggered sensor occurrences, and found item success rate will be partially implemented, as analytics pull from other simulated components. Printable reports, which can be customized to further assist caregivers with tailoring their care plan, will be fully implemented in the prototype. Other data management features, such as item tagging data (which is necessary for identifying an item’s location), will be partially implemented. An image of the item to be found can also be uploaded to the application for ease of identification in later item-finding processes.

2.3 External Interfaces

The Thought Locker Prototype will provide hardware, software, and user interfaces for patients and caregivers to interact with the iOS and Android components of the application. React Native, Google Maps, and Google Calendar APIs will also be used to enhance functional capabilities. All communication that is conducted within the application will utilize secure protocols.
2.3.1 Hardware Interfaces

The Thought Locker prototype is designed to run on iOS or Android devices. It will specifically run on operating systems of iOS 15 and Android 13 or later. Each device must be able to connect to the Internet, have Bluetooth capabilities, and a working camera for full feature functionality.

2.3.2 Software Interfaces

Thought Locker will utilize multiple third-party software and API components. The database will be implemented using a MySQL database housed on a remote ODU server. Google Calendar API will be utilized for task management and Google Maps API will be utilized to simulate the locations of lost items. React Native will be used as a framework to interface with external APIs and provides libraries to work with Bluetooth devices.

2.3.3 User Interfaces

The Thought Locker mobile application will be accessible through a mobile device, downloadable from the iOS or Android app store. The application will conform according to each user’s individual device resolution. Target resolution for Android and Apple devices will be 720x1280, compared to 1792x828 for Apple devices. The landing page of the application will be the login page, with an option available to register a new patient or caregiver. Once logged in, caregivers will have access to all application features. Patients, on the other hand, will only have access to specific features by default. Patient feature permissions can be customized according to each caregiver-patient relationship. In terms of input/output devices, the user will primarily input data using a standard mobile device keyboard and navigation will be conducted using a touch screen display. The camera of the mobile device will be used for uploading images to the item finding feature.
2.3.4 Communications Protocols and Interfaces

Google Maps and Calendar APIs require the HTTPS communication protocol for data transferred over the Internet and OAuth 2.0 for account authentication. The application’s connection to the MySQL database stored on the ODU server will require the FortiClient VPN.

[This space is intentionally left blank.]