Lab 2 – Thought Locker Prototype Product Specification

Nathan Greaney

Old Dominion University

CS411W

Professor James Brunelle

22 October 2023

Version 1 (Draft)
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 13
  2.2 Prototype Functional Description 14
  2.3 External Interfaces 17
    2.3.1 Hardware Interfaces 17
    2.3.2 Software Interfaces 18
    2.3.3 User Interfaces 18
    2.3.4 Communications Protocols and Interfaces 18

List of Figures

Figure 1: Thought Locker Prototype Major Functional Components 14

List of Tables

Table 1: Thought Locker Feature Description and Prototype Implementation 15
1 Introduction

Dementia is a progressive neurological disorder impacting cognitive functions like memory, language, and reasoning. Its prevalence is rising as the global population ages, with 1 in 10 Americans aged 70 and older affected by dementia, 50.4% being mild and 30.3% moderate cases ("Fact sheet," n.d.; "Half," n.d.). This increase has significant implications for healthcare and society, necessitating the understanding of contributing factors and strategies for prevention, early diagnosis, and enhanced care. Dementia progresses through stages, from mild to severe, with symptoms worsening as it advances. Identifying these stages is crucial for tailored care plans focusing on symptom management, healthy living, and quality of life (Alzheimer's Society, n.d.). Although there's no cure for dementia, treatments and management options like medications, lifestyle changes, therapies, and support for patients and families can alleviate symptoms. Thought Locker offers support to individuals with mild to moderate dementia, allowing them to maintain some independence and reduce the burden on caregivers and family members.

1.1 Purpose

The purpose of Thought Locker is to provide comprehensive support and assistance to individuals living with mild to moderate dementia, enabling them to maintain their independence in daily tasks. It offers a range of features, including reminders, item location assistance, and contact support, to enhance the quality of life for patients and reduce the burden on caregivers. Thought Locker also serves as a valuable tool for caregivers by allowing them to monitor their patient's progress and mood changes. The app is designed to be user-friendly, with compatibility for both Android and iOS devices with easy-to-use authentication. Its features, including a calendar, item finders, and location tracking sensors, make it a versatile solution for dementia
care. Furthermore, the integration of motion sensors and patient activity monitoring offers added security and safety for patients. Thought Locker is highly customizable, tailored to each patient's specific needs, and even allows some degree of patient control. This solution combines both hardware (mobile devices) and software (JavaScript, React, Node.js) components to create a seamless and effective user experience. Thought Locker's use of GitLab for version control, issue tracking, and continuous integration ensures that it remains a reliable and up-to-date resource for dementia care.

1.2 Scope

The scope of Thought Locker is to provide support and assistance to a broad spectrum of stakeholders, including dementia patients, their families, friends, and caregivers. Its primary objectives are to promote independence for dementia patients and reduce the stress experienced by their caregivers. The prototype of Thought Locker serves as a demonstration of core functionalities, offering a glimpse into its potential, although not all real-world features are present in the prototype. Thought Locker encompasses a range of hardware and software requirements, ensuring compatibility with Android and iOS devices and utilizing technologies such as JavaScript, React, Node.js, MySQL, AWS, Jest, Git/GitLab, and more. These components form the foundation of the solution.

The prototype's features encompass essential functionalities such as user account management, task tracking, calendars, item location, and notifications. Some of these features employ simulated data, particularly in areas like item location and contact information. Developing Thought Locker also presents a set of challenges. These include the need for familiarity with design tools, complexities related to React Native, effective usage of GitLab, integrating the database, configuring server setup, interfacing with APIs, and sourcing simulated
data. Addressing these challenges will be essential to bring the full set of features to the real-world application and ensure its effectiveness in promoting independence and reducing stress for dementia patients and their caregivers.
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 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.

[ This space left intentionally blank. ]
1.4 References


clock&hl=en_US&gl=US&pli=1

diagnosis/how-dementia-progresses/middle-stage-dementia


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


[ This space left intentionally blank. ]
1.5 Overview

This overview provides an insight into the product specification for Thought Locker, highlighting the architecture design and functional description of the prototype. The product specification for Thought Locker encompasses a comprehensive understanding of its architecture. This architecture is a blend of essential hardware components required for accessing the Thought Locker system and the accompanying development software that powers the platform. Thought Locker's software is a crucial element of its functionality. It comprises a suite of programming languages, covering both front-end and back-end components. These programming languages are carefully selected to ensure the efficient and seamless operation of the system. The product specification will delve into the functional requirements of Thought Locker, providing a detailed account of what the system is intended to achieve. It will also outline how these requirements are implemented within the prototype, offering a practical understanding of how Thought Locker functions and its potential impact on dementia patients, their families, friends, and caregivers.

2 General Description

The Thought Locker prototype is tailored to provide assistance to individuals living with dementia, their families, friends, and caregivers. Its primary objective is to elevate the quality of life for dementia patients by fostering their independence, thereby alleviating stress for all stakeholders. While the prototype may not encompass all the features intended for the real-world application, it effectively serves as a demonstration of its potential, highlighting essential functionalities that empower dementia patients and offer reassurance to caregivers.
2.1 Prototype Architecture Description

The Thought Locker prototype is designed to cater to two main user types: Patients and Caregivers, with the goal of providing support and assistance to individuals with dementia. The prototype application is developed to run on Android and iOS platforms, ensuring accessibility to a wide user base. Thought Locker utilizes AWS as the hosting service for its web application. The front-end and back-end components of the system are seamlessly connected via the React Native API, allowing for efficient communication and interaction between the different parts of the application. The Thought Locker prototype relies on the MYSQL database to store and manage user information, profiles, and data history. This database plays a pivotal role in ensuring the integrity and accessibility of user-related data. The prototype leverages React Native APIs for connecting the Thought Locker database to its server, enabling the exchange of information and data between the two components. Additionally, the Google Maps API is employed by the Thought Locker prototype to generate location-based data, enhancing the functionality of the application by providing valuable location-related information to users. The prototype architecture is illustrated in Figure 1.
2.2 Prototype Functional Description

The Thought Locker prototype is designed to help people with dementia, as well as their caregivers and family. It has various functions, such as user registration, login, and profile management. Users can see their calendar and schedule tasks, marking them as done if needed. They can also find their belongings using Bluetooth and pictures, contact others easily, and access emergency contacts quickly. The app can send alerts if there's a problem and keeps caregivers updated with notifications. It also logs user actions, and there are analytics to understand how the app is used. All these features aim to make life better for those with dementia and their caregivers. The features in Thought Locker, and how they will be implemented, are illustrated in Table 1.

[ This space left intentionally blank. ]
### 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: Location data will be</td>
</tr>
<tr>
<td></td>
<td>enabling location-based services</td>
<td>simulated</td>
</tr>
<tr>
<td><strong>USER INTERFACE</strong></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 Verification</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 Verification</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: Sensor data will be</td>
</tr>
<tr>
<td></td>
<td>assess changes to item locations in the environment</td>
<td>simulated</td>
</tr>
<tr>
<td>Contact Center</td>
<td>Manages user interactions across various communication channels, such as</td>
<td>Partially Implemented: Contact data will be</td>
</tr>
<tr>
<td></td>
<td>phone, email, and text</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: Contact data will be</td>
</tr>
<tr>
<td></td>
<td>typically for medical, safety, or personal reasons</td>
<td>simulated</td>
</tr>
<tr>
<td>Notifications</td>
<td>Alerts delivered through the application to inform users about important</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>information, events, or updates</td>
<td></td>
</tr>
<tr>
<td><strong>DATA MANAGEMENT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Analytics</td>
<td>Collecting and analyzing data about the behavior and interactions of</td>
<td>Partially Implemented: Analytics will be</td>
</tr>
<tr>
<td></td>
<td>patients with the application, aiding in decision-making and patient care</td>
<td>simulated</td>
</tr>
<tr>
<td>Reports</td>
<td>Structured documents that provide information, analysis, and findings on</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>specific user defined topics</td>
<td></td>
</tr>
<tr>
<td>Item Tagging</td>
<td>Assign key metadata values to items, enabling efficient categorization and</td>
<td>Partially Implemented: Geotag data will be</td>
</tr>
<tr>
<td></td>
<td>retrieval within the database</td>
<td>simulated</td>
</tr>
<tr>
<td>Images</td>
<td>Upload images of tracked items, which must correspond to items in the</td>
<td>Fully Implemented</td>
</tr>
<tr>
<td></td>
<td>database</td>
<td></td>
</tr>
</tbody>
</table>
The Thought Locker prototype provides a comprehensive array of functional components designed to support individuals with dementia, their caregivers, and enhance the overall user experience. These major functional components include: User Account Management, User Interface, and Data Management.

In Account Management, Thought Locker ensures secure user registration, distinguishing between patients and caregivers. During registration, a user profile is created and seamlessly linked to the database, facilitating effective user data management. Users can securely authenticate and log into the application by using the username and password established during the registration process, ensuring their information is safeguarded. The application allows users to efficiently manage their profiles, offering options to update, delete, and edit permissions, providing control over their personal information.

In User Interface, Thought Locker features a calendar function, allowing users to access and organize events and appointments, aiding in task scheduling and overall time management. Users can easily view their assigned tasks, mark them as completed, and, with the appropriate permissions, modify or delete tasks, enhancing the management of daily responsibilities. Thought Locker assists users in locating their items using Bluetooth tracking and audio cues, with additional visual support from pictures, making it simpler to find personal belongings. The application provides a platform for users to connect with pre-programmed contact numbers, ensuring quick and easy communication with friends, family, and caregivers. Users can also rapidly cycle through a set of emergency contacts with the touch of a button, offering immediate access to help in critical situations.

In Data Management, Thought Locker enables users to configure and receive monitoring alerts through pre-placed sensors, contributing to enhanced safety and security. The application
sends notifications to caregivers, keeping them informed about important updates and events related to the patient's well-being. Users, primarily caregivers, can access logs of application events, offering insight into user activities and interactions, facilitating comprehensive monitoring. Thought Locker also provides user analytics with the option to download, delivering valuable insights into various aspects of user engagement and behavior, including login data, task management, item tracking, contact logs, and monitoring alerts, enabling a deeper understanding of application usage. These functional components collectively aim to provide comprehensive support and improved well-being for individuals with dementia and their caregivers.

2.3 External Interfaces

Thought Locker provides versatile interfaces, including mobile devices with Android and iOS support, and web browsers for Linux, Windows, and Mac OS. The software interfaces encompass GitLab for version control, MySQL for data management, AWS for hosting, and Android Studio for Android development. Additionally, the application employs communication protocols like IEEE 802.11b-ax for Wi-Fi, HTTPS for secure data transfer, and OAuth 2.0 for user authentication. These interfaces and protocols collectively enhance accessibility, data security, and the user experience for individuals with dementia and their caregivers.

2.3.1 Hardware Interfaces

Thought Locker is compatible with Android devices (Android 11 or higher) and iOS devices (iOS 15 or higher), making it accessible on a wide range of mobile devices. Additionally, it can be accessed on computers running Linux, Windows, or Mac OS through modern web browsers, ensuring flexibility and convenience for users on different platforms.
2.3.2 Software Interfaces

Thought Locker interfaces with key software components to enhance its functionality and reliability. It uses GitLab for version control, MySQL for data management, AWS for hosting the web application, and Android Studio for developing the Android version. These interfaces are essential for efficient development, data management, hosting, and compatibility with Android devices, contributing to the overall performance of the system.

2.3.3 User Interfaces

Thought Locker offers a user-friendly interface accessible via a mobile device with an internet connection. This interface is designed to provide a seamless and convenient user experience, allowing individuals to access and utilize the Thought Locker application on their smartphones or tablets while connected to the internet. The mobile interface is optimized for ease of use, starting with a login screen followed by a home page, which provides the user with links to the major functions of the application.

2.3.4 Communications Protocols and Interfaces

Thought Locker utilizes secure and efficient communications protocols and interfaces to ensure a reliable and safe user experience. It employs IEEE 802.11b-ax for robust Wi-Fi connections, HTTPS for secure data transmission, and OAuth 2.0 for user authentication and access control, collectively contributing to data security and user privacy.