Lab 1 – Thought Locker Product Description

Deshawn Bailey, Olivia Gee, Nathan Greaney, Jerome Palmer, Ty Schiffer, Ryan Wilkinson

Old Dominion University

CS411W

Professor James Brunelle

8 September 2023

Version 2
LAB 1 – THOUGHT LOCKER PRODUCT DESCRIPTION

Table of Contents

1 Introduction 3
2 Product Description 4
   2.1 Key Product Features and Capabilities 4
   2.2 Major Components (Hardware/Software) 5
3 Identification of Case Study 7
4 Product Prototype Description 9
   4.1 Prototype Architecture (Hardware/Software) 9
   4.2 Prototype Features and Capabilities 10
   4.3 Prototype Development Challenges 12
5 Glossary 13
6 References 17

List of Figures

Figure 1: Thought Locker Major Functional Components Diagram 5
Figure 2: Thought Locker Prototype Major Functional Components Diagram 10

List of Tables

Table 1: Thought Locker RWP vs. Prototype Features 9
1 Introduction

- One in ten Americans ages 65 and older have some form of dementia.
- 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 need very little assistance in daily living activities.
- Mild dementia characteristics include changes in mood changes, memory problems, and experience difficulty effectively planning and thinking things through.
- Individuals with moderate dementia may now need frequent reminders and some assistance to wash and dress.
- Moderate dementia characteristics include development of paranoia, intense symptoms of anxiety and depression, and memory problems worsen.
- If a dementia patient is in the mild or moderate category they would prefer to maintain some of their independence.
- As dementia symptoms worsen, assistance is required from either a family member or outside source.
- Care becomes stressful for the caregiver/family member because of the required constant attention.
- The number of individuals diagnosed with dementia is steadily increasing over the years.
- Thought Locker is a mobile assistant designed to help dementia patients maintain their independence through reminders, item location, monitoring, and analytics that will be provided to the present caregiver.
2 Product Description

2.1 Key Product Features and Capabilities

- Compatibility
  - Can be used on smartphones, tablets, and other devices on Android and iOS operating systems.

- Authentication
  - Users will have to sign up for an account using their email address.

- Calendar
  - Users will be able to schedule reminders.
  - Customize settings for reminders and notifications.

- Item Finders
  - Users will be able to locate commonly misplaced items
  - Integrate sensors to track item location.

- Sensors
  - Users will have motion sensors to detect activity in the home.

- Notifications
  - Users will be provided with daily reminders and notifications.

- Patient Monitoring
  - Ability for family members or caregivers to monitor patient activities.
  - Provides peace of mind for family members and caregivers.

- Customizable Settings
  - Family members or caregivers can customize the app for each patient’s needs.
  - Patients can control some settings to maintain a sense of independence.
2.2 Major Components (Hardware/Software)

Figure 1

*Thought Locker Major Functional Components Diagram*

- **Hardware**
  - Android or Apple mobile device with Internet Access
  - Application Server
  - Database Server

- **Software**
  - Backend Server- AWS
  - Databases - MySQL
  - Programming Language - Javascript(React/Node.js)
  - Testing - Jest
  - Repository - GitLab
  - Version Control - GitLab
○ Issue Tracking - Jira
○ CI/CD - GitLab
○ Containerization - Docker

[ This space left intentionally blank. ]
3 Identification of Case Study

- Who is Thought Locker for?
  - Individuals with mild to moderate dementia
  - Caregivers / family members of individuals with mild to moderate dementia

- What will Thought Locker be used for?
  - Granting caregivers and patients an easy method to locate lost or misplaced items.
  - Retaining analytics for caregivers to visualize and monitor patient habits, such as the number of times an item has been lost or the number of times a door has been left open
  - Reminding patients to take their medication or inform them of upcoming appointments
  - Providing patients with the opportunity to immediately contact a caregiver with an urgent need
  - Offering patients and family members a more cost-effective option than hiring a full-time caregiver

- Who else might benefit from Thought Locker?
  - Medical facilities
  - Insurance companies / Medicaid
  - Alzheimer’s research groups

- What user cases will be demonstrated in the prototype?
  - Individuals with dementia
    - 6 users (3 mild, 3 moderate) mocked up
    - Item finding
- Scheduling
- Contacting

  - Caregivers / family members
    - 3 users mocked up
    - Monitoring
    - Providing help
    - Analyzing trends

[ This space left intentionally 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.

Table 1

Thought Locker RWP vs. Prototype Features

<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></th>
<th></th>
<th></th>
<th></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></th>
<th></th>
<th></th>
<th></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>

4.1 Prototype Architecture (Hardware/Software)

Hardware

- Mobile phone or tablet operated by Android/Apple
- Desktop or laptop operated by Linux, MacOS or Windows
Software

- Languages: Javascript, React, Node.js
- Database: MySQL
- Back-End Server: Amazon Web Services (AWS)
- Testing: Jest
- API: Reactive Native, Google Maps Geolocation API
- CI/CD, Version Control: Git/GitLab

**Figure 2**

*Thought Locker Prototype Major Functional Components Diagram*

4.2 **Prototype Features and Capabilities**

- Some aspects of the prototype will be fully implemented, while others will have partial implementation or be eliminated entirely.
- Full implementation of user account management (except simulated location information)
  - Account creation/deletion
- Profile/preference management
- Login capability

- Full implementation of some user interface features
  - Task management: including addition, deletion and scheduling of tasks
  - Calendar: Can add appointments and tasks for the patient to accomplish throughout the day
  - Addition and deletion of items often lost by the patient
  - In-app notifications

- Partial implementation of features that require the use of external data sources. This is because it is not important from a prototype perspective; the prototype is meant to showcase in-app functionalities, not necessary to have real-world data.
  - Item locator: Item location data will be simulated (Random item locations will be generated on a map for purposes of showcasing the item locator feature).
  - Contact Data: Will use simulated contact information; parameters for adding/deleting contacts (name, phone number, etc.) will be implemented.
    - Emergency contact features will utilize simulated contact information, but the process of calling and notifying an emergency contact will be fully implemented.

- Remote access will be eliminated as it does not directly address the problems those with dementia face; it is instead more of an embellishment that can be added later.

- Data Management: Some features fully implemented, others partially implemented.
  - User reports and item tag data will be simulated, as additional testing with the target market is needed before we can generate reports more accurately.
○ Formatted reports and images used to track items will be fully implemented, as they will showcase how caregivers can learn from patient activity to better tailor their care plan.

4.3 Prototype Development Challenges

● Resource Limitations
  ○ Limited time and labor can impact the development process.

● Technical Feasibility
  ○ Implementation challenges may appear for certain features.

● Hardware/software Compatibility
  ○ Compatibility issues can occur, especially concerning specific hardware and software configurations.
  ○ Demonstrating the desired functionality can be challenging.

● Testing and Bug Fixing
  ○ Identification and resolution of bugs and issues during development.
  ○ Thorough testing is essential to ensure the prototype functions correctly.

● Scalability
  ○ Ensuring the prototype can handle real-world problems and scales effectively is necessary.

● Prototyping Tools and Platforms
  ○ Choosing the right tools and platform for the prototype is important during the development process.
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.

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 left intentionally blank. ]
6 References


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


18816