Requirements Specification

DAC Project

CS 448
Austin Bos, Aaron Saucedo, Ivan Nieto

Requirements Specifications 001 for notetaking application requested by the Army Data and Analysis Center

Date

02/29/2020

Table of Contents

Introduction

The goal of this project is to develop an application that aids the CCDC-DAC analysts to collaborate and incorporate each other's work when constructing reports on the cyber vulnerabilities found for an investigated system or location. This application will not only merge the findings of the analysts into a single report but also help organize the investigations by allowing the lead analyst to assign tasks to the other analysts.

1.1 Purpose of Product

The purpose of this application is to aid the CCDC-DAC analysts to better communicate and collaborate. With more communication the CCDC-DAC team can better identify critical cyber vulnerabilities and provide mitigation recommendations. This application will help the analysts store and merge their findings with each other and kickstart the creation of a final report.

1.2 Scope of Product

For our project we will be building an electron application that will consist of three parts. First, the frontend of the application, created using HTML, CSS and JavaScript with the help of the React library will allow the user to edit, view or input their findings and save them to their local database as well as allowing the creation of tasks and delegation of these tasks to other users. The second part of the application will be the database which will be responsible for storing all the user's information. The final part of the application will be the backend which will be written in python. This backend will serve as the connection between the frontend and the database and will also handle all work associated with syncing the user's local database with another user's application when prompted to by the user from the frontend.

- consist of database and networking
- should allow assigning task and findings
- networking with connecting members with each other

1.3 Acronyms, Abbreviations, Definitions

- CCDC-DAC – Combat Capabilities Development Command Data and Analysis Center

-Frontend - The applications graphical user interface where the user will be able to access all functionality for the application.

-Backend - Segmented part of application handling application tasks not seen by the users.

-DOD - Department of Defense

-CEAD - Cyber Experimentation & Analysis Division

-SME - Subject Matter Expert

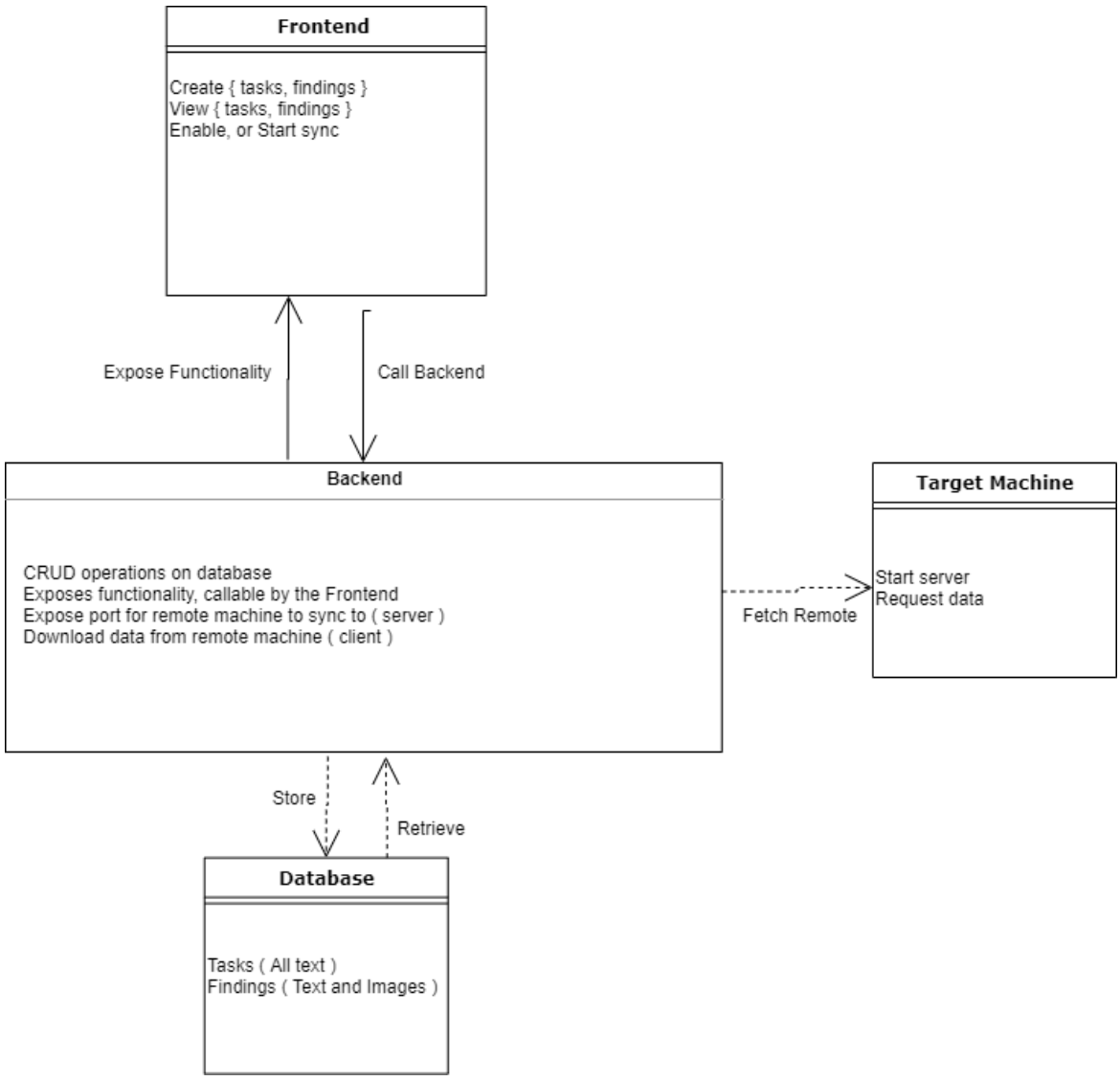
-Python - A programming language with an uncomplicated application programming interface facilitating a connection with the SQLite database.

-Electron - An application building framework that simplifies the creation of desktop graphical user interfaces using web technologies and programing languages.
-React - A JavaScript library that helps in the creation of large web-based applications.

General Description of Product

Overview of the product and its environments

2.1 Domain Model



The Domain Model above shows how the application will be segmented into its three main pieces. The Frontend will be the main point of contact with the user and will communicate

with the backend of the system. The backend will consist of scripts that will communicate with the database which will be triggered by the user on the Frontend. The user machine above represents another user that would be the target of the sync mechanism. The database will be accessed only by the backend and will contain all the persistent data.

2.2 Context of Product

This application will be used on portable government issue laptops that are expected to operate in a variety of environments. It will primarily be run on a Kali Linux operating system and will not have access to external resources. The application will only communicate with outside resources when syncing its data with another machine through a local area network.

2.3 User Characteristics

Our users will be CEAD analysts who all have backgrounds in computing. We do not expect our users to install any dependencies for our application other than the application itself. We will handle two types of users, a lead analyst and a normal analyst. Both titles will be interchangeable, and the only difference will be that whoever is designated as lead analyst will be able to sync everyone else's databases to their machine.

-there are 2 roles
-Members are assigned task, do pen testing and report findings and sync to lead
-Lead assigns task to members and syncs to all members of a team

2.4 Constraints

Our technical constraints will pertain to the systems that our application is intended to work in, and it's intended use. This means we will be constrained to developing our application for full compatibility with Kali Linux. Another constraint will be the lack of dependencies allowed for the application. No dependencies must be required for the application to work as intended; all requirements must come packaged with the application. The application must not communicate with any outside system other than when performing a sync with another application. Regarding the sync functionality the application must encrypt all data being transferred through the local area network and all information being handled by the application must not be shared with any outside systems.

-Must work on Kali Linux operating systems
-Must run a security protocol within the network
-Information cannot leak outside the network
-Must be executable from a single file (don't need to download dependencies)

2.5 Assumptions and Dependencies

This application will be required to operate without external software dependencies therefore all dependencies must come packaged with the application.

Specific Requirements

System Specifications

3.1 External Interface Requirements

3.1.1 User Interface

The user interface will provide the user the ability to view, edit and create findings and tasks. It will also allow the user to begin the synchronization process with another machine. The user interface will be a GUI with the form of a website. The GUI will consist of multiple pages that will display the data from the user's local database and allow the user to enter new findings and tasks into their local database. All the user's interactions with the application will be through the user interface.

3.1.2 Communications Interface

For the communication interface, each computer, except, the team lead, will open a gRPC server, from which the team lead can use to access, and download all their data to their machine. The gRPC server will be closed by default, and the user will be able to open the server on demand.

3.2 Functional Requirements

3.2.1 User Interface

The user interface will provide two main functions. First it will allow the user to view, edit and create tasks and findings. The user will enter their findings into a text box and will have the ability to add images if desired. Once the user has entered their finding, they will then save their changes to the database by clicking a button. The second function of the frontend will be to begin the process of syncing the user's data with another user. This will be initiated by inputting the target machines IP address and then selecting what data to sync. All user input for the application will be handled by the frontend.

3.2.2 Database

We will use a database to store all the data for our application. To allow the application to successfully sync its data with another user we will be using the user's unique username as part of the primary key of all the tables in the database. The tables in the database will store tasks,

findings, usernames, and other relevant data of the application. All access to the database will be handled by the backend.

3.2.3 Backend

The backend will serve all the data handling needs. This will include database CRUD operations, syncing to team lead, and automated report generation. The backend will expose functionality in such a way, that the front end will be able to utilize its functionality. The database operations will be wrappers around needed database functionality, offering a simple interface to the front end to store, and retrieve data. The backend will also offer up functionality that will allow client/server communication for syncing data. An input may be given, such as machine information, and a subset of data that can be either offered up to sync or pulled from a remote machine. The last backend feature that will be available is going to be for report generation. For report generation, it will accept an input of data, and result in an editable .docx file.

3.3 Performance Requirements

The application must also encrypt any information being sent over the local area network and must not lose any of the information when syncing. Data integrity and security must be the top priority in the syncing process. Data integrity at rest for no corruption or unintentional changes/deletion.

3.4 Design Constraints

The application needs to establish connections inside a local network and will be layered, so we will need to implement a client-server design combined with a layered design. The application will have an UI (presentation), containing services to be run for it to allow editing, uploading and synching database information (application). It will also have business rules to comply with as leads assign tasks while members cannot. The application also cannot connect outside the local network (Business layer) and must be completely isolated from outside resources. The last layer will be the database that will be implemented inside the application (Data access).

3.5 Quality Requirements

The application must run consistently and dependably every time it is run. The data entry and sync functionality should be easy to find and interact with.