Access Control Software

Hannah Protzman, Peyton Mizera, Kyle Warren, Cassie Baldus, Jeremiah White

NWMSU Access Control Committee

**Initial Project Proposal**

**Title:** Access Control Software

**Team members:** Hannah Protzman, Cassie Baldus, Kyle Warren, Peyton Mizera, Jeremiah White

**Client:** Facility Services Access Control

**Introduction:** One of our group members works with the client and noticed that their key tracking system is currently all on paper. Our goal is to streamline their processes by making it all digital.

**Objectives:**

1. Take the client’s key-tracking system, which is currently all on paper, and digitalize it
2. Create a digital authorization process – set this up for client instead of employees having to sign paper
3. Create a database of all employees that have keys and where they have access to

**List of Target Customers/Users:** Employees of Northwest Missouri State University and Access Control Managers

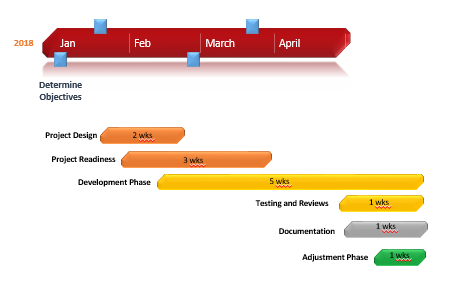
**Value of the project:** Provides value because it makes things more efficient and easier for the employees. It is streamlining the process by taking something that is outdated and unorganized and making it more resourceful and proficient.

**Features:**

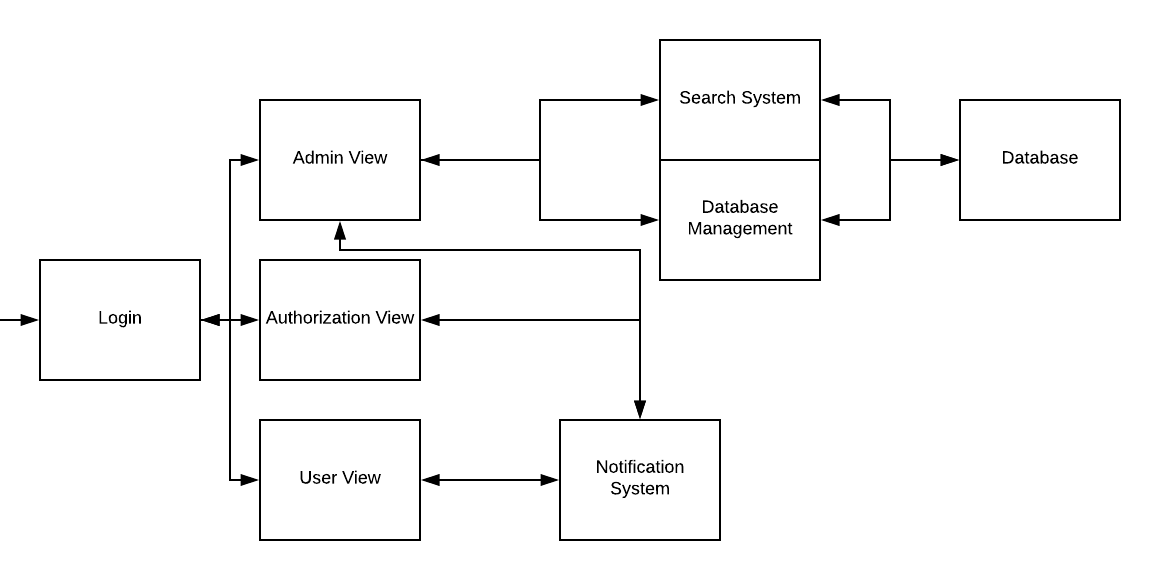
* Easily searchable database that holds employees’ names, keys, and where they can access
* Easy-to use-request system for keys
* Digital authorization process

**Tools & Resources:**

* MongoDB system
* HTML/CSS/JavaScript to create webpage

1. **Project Management**
2. **Schedule:** 
   1. Week 1 –
   2. Meet with client
   3. Requirements from client
   4. Abstraction and design in class on Thursday
   5. Week 2 –
   6. Basic design and mock-up
   7. 
3. **Tasks:**
   1. HTML/CSS/Website Design: Hannah Protzman, Cassie Baldus
      1. Request page, authorizer page, access control page
   2. Database Management/Ownership: Kyle Warren, Peyton Mizera, Jeremiah White
   3. Back-end Development: Kyle Warren, Peyton Mizera, Jeremiah White
4. **Weekly Meeting:** On Thursdays at 6pm
5. **Risk Management**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date Recognized** | **Risk** | **Description** | **Type** | **Probability of occurrence** | **Severity**  **(1-3)** | **Action & Response** | **Status** | **Priority** |
| 1.30.18 | Security | Someone who is not an admin could say that they are | Technical | Low | 3 | Police | N/A | N/A |
| 1.30.18 | Keys become obsolete | What is the university decides to stop using keys altogether? | Project | Low-Medium  33.33333% | 2 | Have to change the system completely | N/A | N/A |
| 1.30.18 | Misplacement | The keys getting into the wrong hands | Technical | Low | 3 | Replace the key and do our best to make sure to find the lost key | N/A | N/A |
| 1.30.18 | Skill set | There are areas in which we may not necessarily know what we are doing | Project | High | 2 | We will learn a lot about this project! | N/A | N/A |

1. **Requirements**
2. List of Functional Requirements
   1. Functional Requirement 01
      1. User Requirement
         1. REQ#1
         2. TITLE: Login Page
         3. DESC: Application needs a login page to authorize user
         4. RAT: To make sure the user has specific authorization to view certain pages
         5. DEP: Depends on authorization system that is already set up by campus
      2. System Requirement: Use an HTML/JavaScript form to create login page
   2. Functional Requirement 02
      1. User Requirement
         1. REQ#2
         2. TITLE: User Authorization/Admin Pages
         3. DESC: Interface between the different types of people who will be using the system
         4. RAT: To ensure safety and organize the system of access
         5. DEP: REQ#1
      2. System Requirement: Using HMTL to create database on who can authorize what
   3. Function Requirement 03
      1. User Requirement
         1. REQ#3
         2. TITLE: Notification System
         3. DESC: System to notify admin when key requests are sent in, when key is ready to be sent, etc.
         4. RAT: Helps system run more efficiently
      2. System Requirement: Use HTML and JavaScript to send an email to admin whenever a form is submitted
   4. Functional Requirement 04
      1. User Requirement
         1. REQ#4
         2. TITLE: Request Form
         3. DESC: A form for users to fill out in order to request a key
         4. RAT: Helps the admin keep track of what requests has been filled, makes requesting very easy for users
      2. System Requirements: Use HTML to create a simple and easy to use form
   5. Functional Requirement 05
      1. User Requirement
         1. REQ#5
         2. TITLE: Database
         3. DESC: A database to keep basic information
         4. RAT: Will be a simple database to keep track of who has what keys, very simple, not a lot of complexity for ease of use
      2. System Requirements: We will be using MongoDB and SQL to create and manage our database
   6. Non Functional Requirements
      1. Security: Make sure that only those who are authorized to access rooms have the keys. This could be very important to keep the school safe.
      2. Functional: The system has to actual work and be easy for the users to understand
      3. Quality: The system should be efficient and help make the clients’ lives better
      4. Scalability: We want to be able to keep track of a large database across all of campus.
   7. Open Issues
      1. How to handle if a user loses a key. We need to think about how to handle this issue with the client. We may not be able to get to it before the end of the semester.
      2. Integration with the on-campus CMMS system. We would love to be able to integrate with this system to make things easier for the client, but we probably will not be able to get to it in time.
      3. How can we make the system secure? We may need to learn more about this process in order to keep things secure. It will be on Northwest’s servers so it may not be a huge issue.
   8. Client meeting
      1. Date/Location: Thursday, February 1st, AC office
      2. Attendees: Brittany (head of Access Control Committee), Jeremiah White
      3. Discussion: Discussed the general project and asked about their current key facilitating process. We wanted to know about how long it currently takes them, so that we can make sure our process will be more efficient.
      4. Actions:
         1. Database design: Next 2 weeks, Jeremiah White
         2. Detailed, functional layout, Next 2 weeks, Kyle Warren, Peyton Mizera
         3. Need to figure out how we are going to run the website before it is on the servers, Next 2 weeks, All members
         4. Construct a mock-up of what we want the system to look like, Next week, Hannah Protzman, Cassie Baldus
         5. Get a handle on how to use React, Next week, All members
      5. Next meeting: Sometime during next week
3. **UML Architecture**

Our project will consist of the following views:

* Database Management
  + This is for management of the database system
* Database
  + This is where all the information about keys and people and rooms will be stored
* Notification System
  + For the person fulfilling key requests to notify the requester and authorizer that a key has been requested
* Search System
  + For users to be able to query the database to search for keys
* Admin View
  + Just a few accounts who would be able to control anything
* Authorization View
  + People who decide what key goes to whom and what rooms they have access to
* User View
  + View for normal users, the employees requesting the keys
* Login
  + This is the first page the user will see, where they can enter username and password

These are based on user/system requirements.

The first view will be the login page. A user will sign in with username and password.