
PROJECTS REQUESTS APPLICATION REPORT

Prepared for Abdullah Kadri and David Lindsay in partial
fulfillment of the requirements for ENG4003 / ENL4003

Wenbo Peng
Michael Botelho
Rani Khaddage
Frank Ngumanenimana
Anh Duc Ngo

Dream Development
2021-12-10

Industrial Supervisor: Christopher Hahn

Faculty Supervisor: Abdullah Kadri



AUTHORSHIP

Project Responsibilities:

Wenbo Peng – Team lead

Frank Ngumanenimana – Front-end lead

Michael Botelho – Developer

Rani Khaddage – Developer

Anh Duc Ngo – Developer

Writing Responsibilities:

Michael Botelho – Lead Writer

Rani Khaddage – Lead Writer

SIGNED AGREEMENT

We, Wenbo Peng, Michael Botelho, Frank Ngumanenimana, Rani Khaddage, and Anh Duc Ngo confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

ABSTRACT

This report describes the application created by Dream Development for Christopher Hahn (the client) that stores and collects project request applications on their behalf. It details the design choices, technical specifications, future development by another team, and recommendations by our team. The application is developed to be hosted and managed by ITS in Algonquins' Microsoft Azure cloud hosting domain.

TABLE OF CONTENTS

Authorship	2
Signed agreement	2
Abstract	3
List of tables and figures	5
Glossary	5
1. Introduction	6
2. Design	7
2.1 Sign-in	8
2.2 Form Submission	9
2.1.1 Form info	10
2.2 Dashboard	10
2.2.1 Project requests	11
2.2.2 Form Creation	12
3. Technical specifications	13
3.1 Authentication Process	14
3.2 MVC design pattern	14
3.3 Database Structure Diagram	15
4. Testing, results, and analysis	16
4.1 Developer Testing	16
4.2 Client Testing	16
5. Conclusions and recommendations	17
5.1 Conclusions	17
5.1.1 Client recommendations after testing	17
5.1.2 Security Requirements	17
5.1.3 Additional Recommendations	18
6. References	18

LIST OF TABLES AND FIGURES

Figure 1 - Page design Diagram	7
Figure 2 - Form password page	8
Figure 3 - Project request submission	9
Figure 4 - Dashboard project requests list	10
Figure 5 - dashboard project request view	11
Figure 6 - Dashboard form view	12
Figure 7 - Database structure diagram	15

GLOSSARY

Term	Definition
Active Directory	A directory service developed by Microsoft for Windows domain networks.
Azure Blob Storage	A feature in Microsoft Azure that lets developers store unstructured data in Microsoft's cloud platform.
Azure Cloud	A cloud computing service operated by Microsoft for application management via Microsoft-managed data.
BRM	Business Relationship Management serves as an essential link between the service provider and one or more business units at Algonquin College.
Dashboard User	Selected chairs and faculty members at Algonquin College who have access to view, modify, and approve of project requests.
Entity Framework	An open-source object-relational mapping framework for .NET applications supported by Microsoft.
Form	Details of a project request, such as why it's worth completing, how long it should take, what resources are dedicated to it, and so on.
ITS	Serves the technical needs of the students and staff of Algonquin College.
Microsoft Identity Framework	Builds applications that allows users and customers to sign in using their Microsoft identities or social accounts.
Project Request	Used to initiate ongoing analysis of a project concept.
(Technology) Stack	
.NET Core	An open-source managed computer software framework for Windows, Linux, and macOS operating systems.

1. INTRODUCTION

As part of a new business requirement, Algonquin must be able to accept and process an estimated 100+ project requests from students and external partners. This is simply not possible with current procedures as the management of project requests has been done manually with information stored in Microsoft Word documents. The solution Dream development has created moves all data handling to the cloud using technologies Algonquin already implements. Dream development is creating a web application that handles the submission, organization, and processing of project requests simplifying the process for students, external partners, and faculty.

The solution outlined in this report outlines our application developed entirely on a Microsoft stack Utilizing .NET Core, Azure Cloud, Azure Blob Storage, Microsoft Identity Platform, and Entity Framework. This report includes the design of the application, technical specifications, future development by another team, and recommendations from our own team.

2. DESIGN

The web application is designed to simplify the process of submitting project requests for students and external partners. Whether submitting, viewing, or modifying requests, there are minimal steps, and the flow of the application is easy to understand.

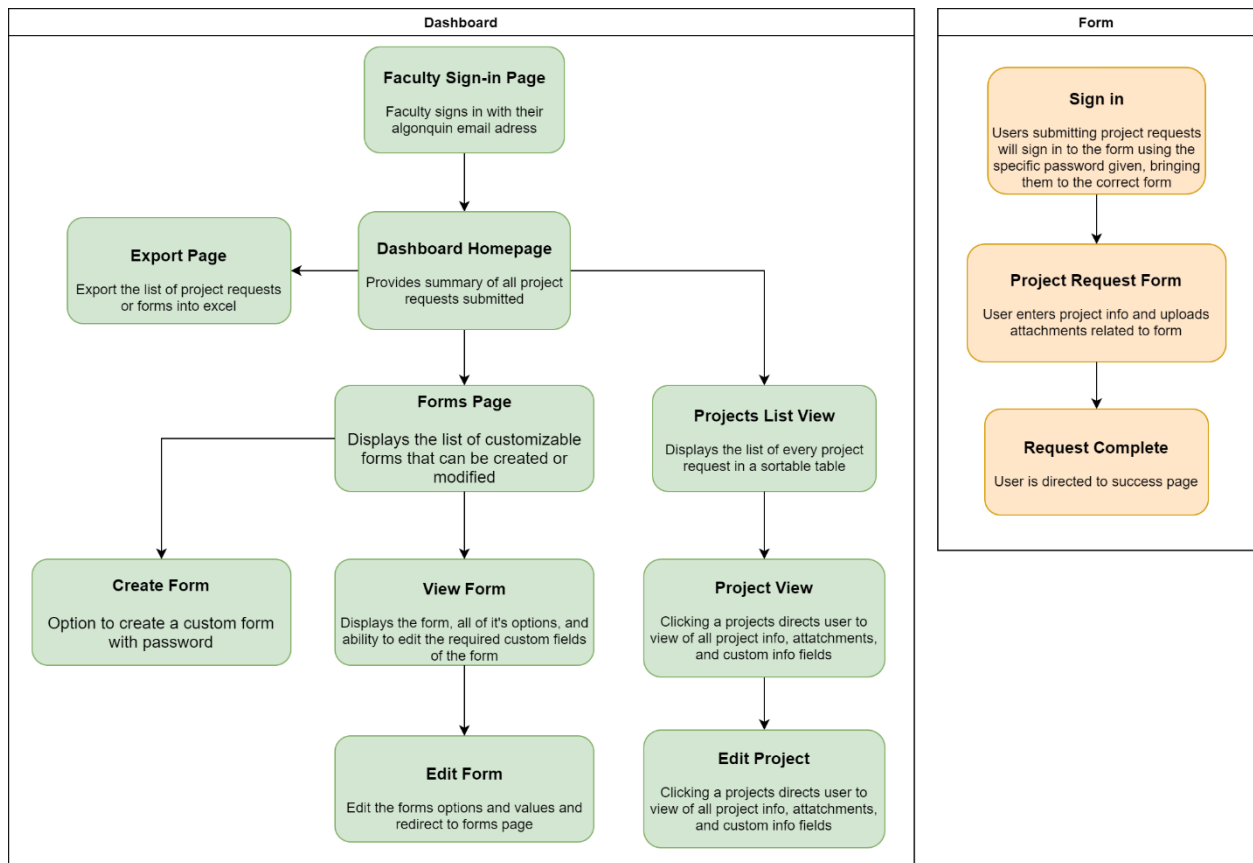
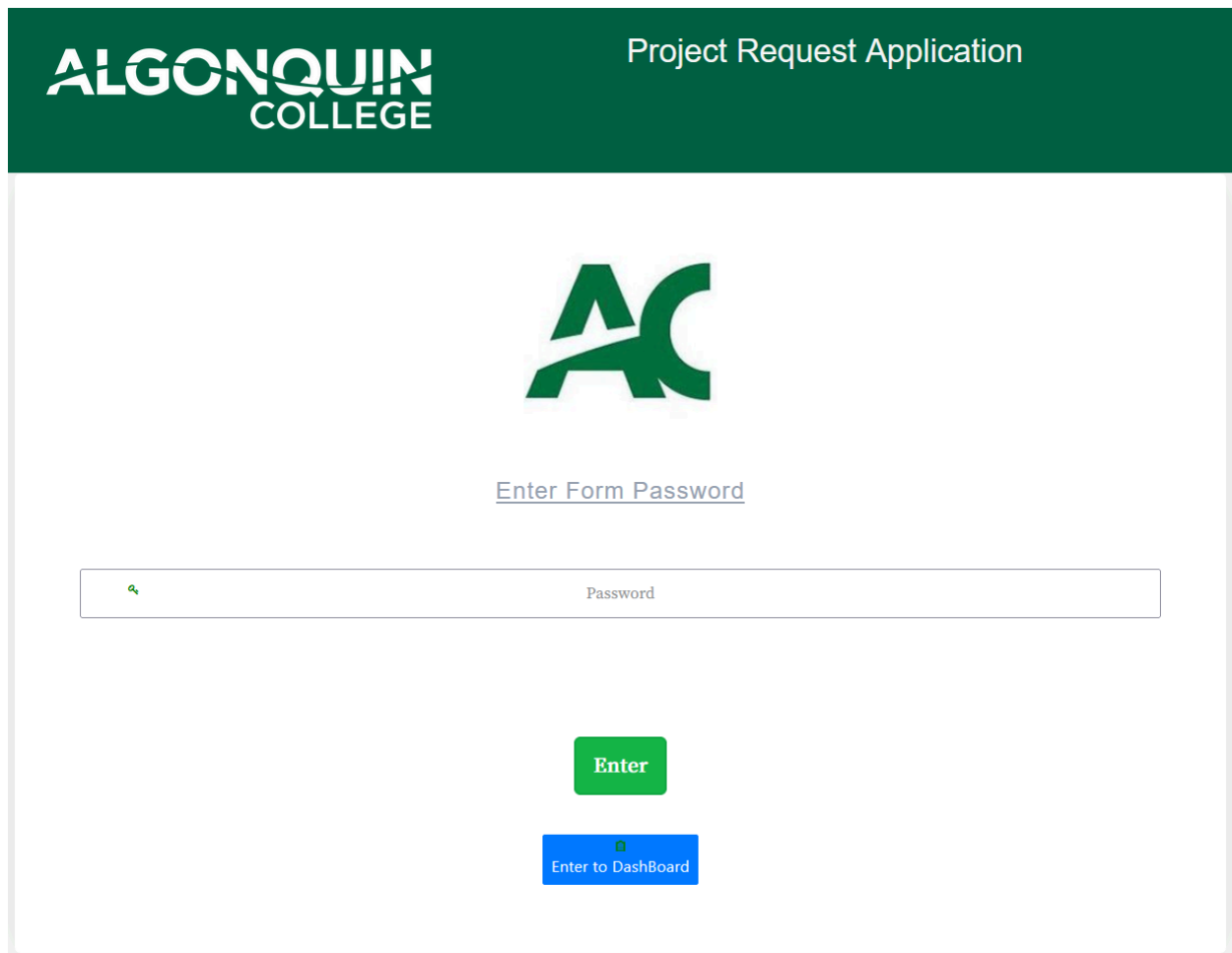


FIGURE 1 - PAGE DESIGN DIAGRAM

2.1 SIGN IN

Both students and external partners are met with a sign in page when first opening the web application. If the user is given a password to a specific form, they can enter it and sign in to access the specific project requests form. Dashboard users can simply log in to the dashboard using their Microsoft account by clicking the “Dashboard Sign-in” Button on the password page.



The screenshot displays the sign-in interface for the Project Request Application. At the top, a dark green header bar contains the Algonquin College logo on the left and the text "Project Request Application" on the right. The main content area is white and features a large green "AC" logo in the center. Below the logo, the text "Enter Form Password" is displayed. A password input field is positioned below this text, containing a small green eye icon on the left and the placeholder text "Password". Underneath the input field are two buttons: a green "Enter" button and a blue "Enter to DashBoard" button, which includes a small Microsoft account icon.

FIGURE 2 - FORM PASSWORD PAGE

2.2 FORM SUBMISSION

The form submission section is where the students and external partners submit the information pertaining to their project request. The form design is based on the original project request document. Since the goal of the application is to simplify the project application process, the online form precisely matches the project request document along with a section for custom required fields determined by the Dashboard user who created the form.

The screenshot displays a web form titled "Project Report Test Form" within a green header bar that also contains the "ALGONQUIN COLLEGE" logo and the text "Project Request Application". Below the header, a subtitle reads "Project Report Test Form" and a descriptive line states: "This is a form to display the design of the project request form page customized by the title above, the description, and the custom fields created by the Dashboard user." The form is organized into two main sections. The first section, "Company or Individual Information", includes a "Project Name*" field, a "Company Name (if applicable)" field, an "Indicate*" section with radio buttons for "For-Profit" and "Not-for Profit", and four individual fields for "First Name:*", "Last Name:*", "Phone:*", and "Email:*". The second section, "Additional Fields", contains two custom fields: "Custom Field One*" (described as a required example) and "Custom Field Two" (described as an optional example).

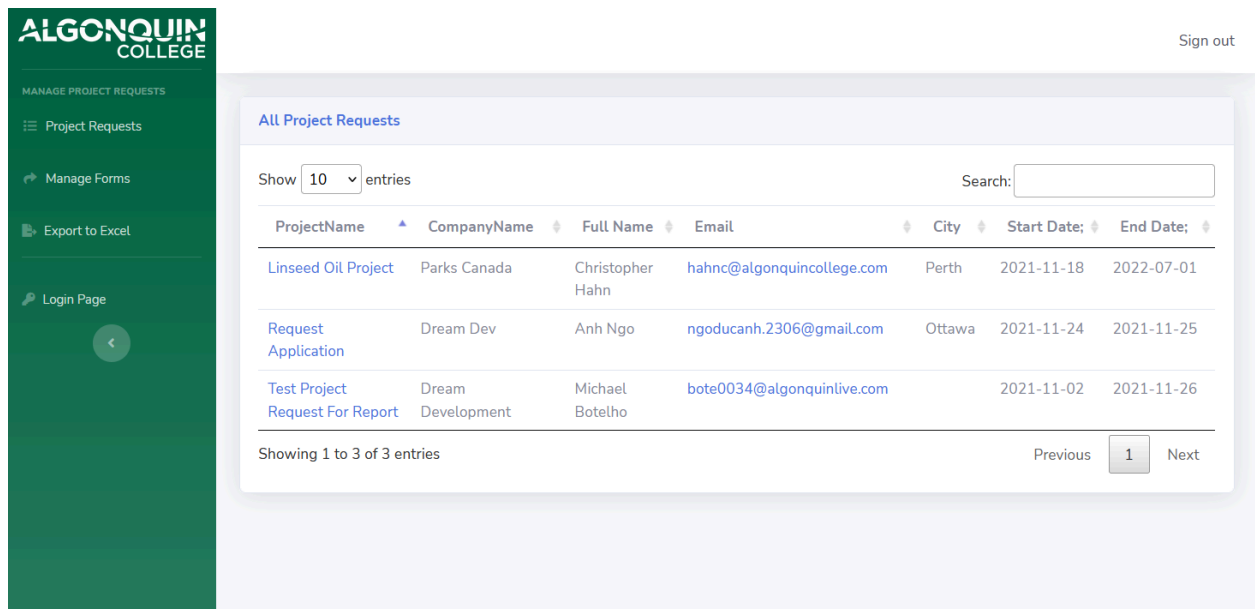
FIGURE 3 - PROJECT REQUEST SUBMISSION

2.1.1 FORM INFO

The project request form includes fields for contact information, project information, collaboration, contribution, and related files as it does on the project request document. The user simply submits the required information and is redirected to a page showing them their project request. As part of a new request from the client, Dashboard users can add custom fields to the forms themselves and decide whether to ask for an address as part of the form.

2.2 DASHBOARD

The dashboard section of the application is where the client and other faculty will view, modify, and approve of project requests. The design of the dashboard was based on the client's needs to efficiently manage hundreds of project requests. A dashboard layout was created for the purpose of viewing large amounts of data.



The screenshot shows the Algonquin College dashboard interface. On the left is a dark green sidebar with the college logo and navigation links: 'MANAGE PROJECT REQUESTS', 'Project Requests', 'Manage Forms', 'Export to Excel', and 'Login Page'. The main content area is titled 'All Project Requests' and features a table of project requests. Above the table are controls for 'Show 10 entries' and a search bar. The table has columns for ProjectName, CompanyName, Full Name, Email, City, Start Date, and End Date. Three entries are visible: 'Linseed Oil Project' by Parks Canada, 'Request Application' by Dream Dev, and 'Test Project Request For Report' by Dream Development. At the bottom of the table area, it says 'Showing 1 to 3 of 3 entries' and includes 'Previous', '1', and 'Next' pagination links. A 'Sign out' link is in the top right corner.

ProjectName	CompanyName	Full Name	Email	City	Start Date	End Date
Linseed Oil Project	Parks Canada	Christopher Hahn	hahnc@algonquincollege.com	Perth	2021-11-18	2022-07-01
Request Application	Dream Dev	Anh Ngo	ngoducanh.2306@gmail.com	Ottawa	2021-11-24	2021-11-25
Test Project Request For Report	Dream Development	Michael Botelho	bote0034@algonquinlive.com		2021-11-02	2021-11-26

FIGURE 4 - DASHBOARD PROJECT REQUESTS LIST

2.2.1 PROJECT REQUESTS

The page for viewing a single project request simply contains all fields pertaining to the project request selected and a section for the dashboard user to add extra information or attachments as required. Since a project request contains such a large collection of information, a single view page is used per project request.

The screenshot displays the 'Project Information' view for a specific project request in the Algonquin College system. The interface includes a green sidebar with navigation options: 'MANAGE PROJECT REQUESTS', 'Project Requests', 'Manage Forms', 'Export to Excel', and 'Login Page'. The main content area is divided into several sections: 'Project Information' with fields for PROJECTNAME, COMPANYNAME, INDICATE, FULL NAME, PHONE NUMBER, EMAIL, START DATE, and END DATE; 'Custom Fields' for additional data; 'Attachments' showing two document files; and 'Additional notes' for user comments. A 'Sign out' link is located in the top right corner.

ALGONQUIN COLLEGE

MANAGE PROJECT REQUESTS

Project Requests

Manage Forms

Export to Excel

Login Page

Sign out

Project Information [Edit] [Delete]

PROJECTNAME
Test Project Request For Report

COMPANYNAME
Dream Development

INDICATE
For-Profit

FULL NAME
Michael Botelho

PHONE NUMBER
5555555555

EMAIL
bote0034@algonquinlive.com

START DATE
2021-11-02

END DATE
2021-11-26

ADDITIONAL INFO
This is where additional project information is added that the student or external partner might believe is necessary. Anything can be put here.

Custom Fields

CUSTOM FIELD ONE
Value for custom field One

CUSTOM FIELD TWO
Another Custom Value

Attachments

2 files selected [Browse ...]

Additional notes [Edit Notes]

These are additional notes that only dashboard users can see.

2.2.2 FORM CREATION

Dashboard users can create and manage forms through the forms section on the dashboard. In this section of the dashboard, they can view a list of all forms, manage each form, and manage custom fields that are unique to each form. In the current state of development, only text fields can be added as custom fields. This feature can be expanded to include other types of fields during future development by other teams.

ALGONQUIN COLLEGE

MANAGE PROJECT REQUESTS

- Project Requests
- Manage Forms
- Export to Excel
- Login Page

Sign out

Project Report Test Form

[Edit](#) [Delete](#)

TITLE
Project Report Test Form

DESCRIPTION
This is a form to display the design of the project request form page customized by the title above, the description, and the custom fields created by the Dashboard user.

PASSWORD
Draft123!!

ADDRESS OPTION
☐

END DATE
11/30/2021

Custom Fields

[+ Add Field](#)

Title	Description	Required	Options
Custom Field One	An example of a custom field that is required	<input checked="" type="checkbox"/>	Edit Delete
Custom Field Two	An example of a custom field that is optional	<input type="checkbox"/>	Save
Custom Field Three	Example	<input checked="" type="checkbox"/>	Edit Delete

3. TECHNICAL SPECIFICATIONS

The application is developed and hosted using an Azure/Microsoft stack to easily integrate with Algonquins current processes. By using the same technology and hosting provider as Algonquin, the project requests application can easily integrate with Algonquins current @algonquincollege domain and be hosted on the same servers that ITS uses [1].

- Development Environment – Visual Studio 2019 (.NET Core 5 MVC)
 - Visual studio provides the tools and technologies necessary to integrate with Microsoft's Azure platform. The process of storing data on the Azure cloud and login management through Microsoft's Identity Framework is simplified and secure.
 - The MVC application design model separates the design, business, and database logic to organize such a complex application. [2]
- Hosting Platform – Microsoft Azure
 - By using Microsoft's Azure platform, the application can be hosted, managed, and funded by Algonquin easily since Algonquin already depends on the Azure platform.
 - The Database, blob storage, and authentication system are all hosted on the Azure platform to ensure the security, scalability, and longevity of the application.
 - Discussions are currently ongoing with ITS for the future management of the application across the long-term including the option of Algonquin themselves hosting and managing the application after completion

3.1 AUTHENTICATION PROCESS

The authentication of faculty is handled through the Microsoft Identity Framework and Azure Active Directories. The team is currently working with BRM, ITS and a Microsoft Collaboration Suite Architect to get the application's Authentication system working with Algonquins Active Directory. This way faculty can securely log into the application through Microsoft itself without giving the development team access to any @algonquincollege passwords. [1] Due to the current workload of ITS, implementation of the Algonquin directory has been pushed to a later date.

3.2 MVC DESIGN PATTERN

By using the MVC design pattern the applications design, business, and database logic are separated and can be modified easily [2]. This design model also allows for easy implementation of Microsoft technologies as middleware such as Authentication, Entity Framework, and Blob storage.

3.3 DATABASE STRUCTURE DIAGRAM

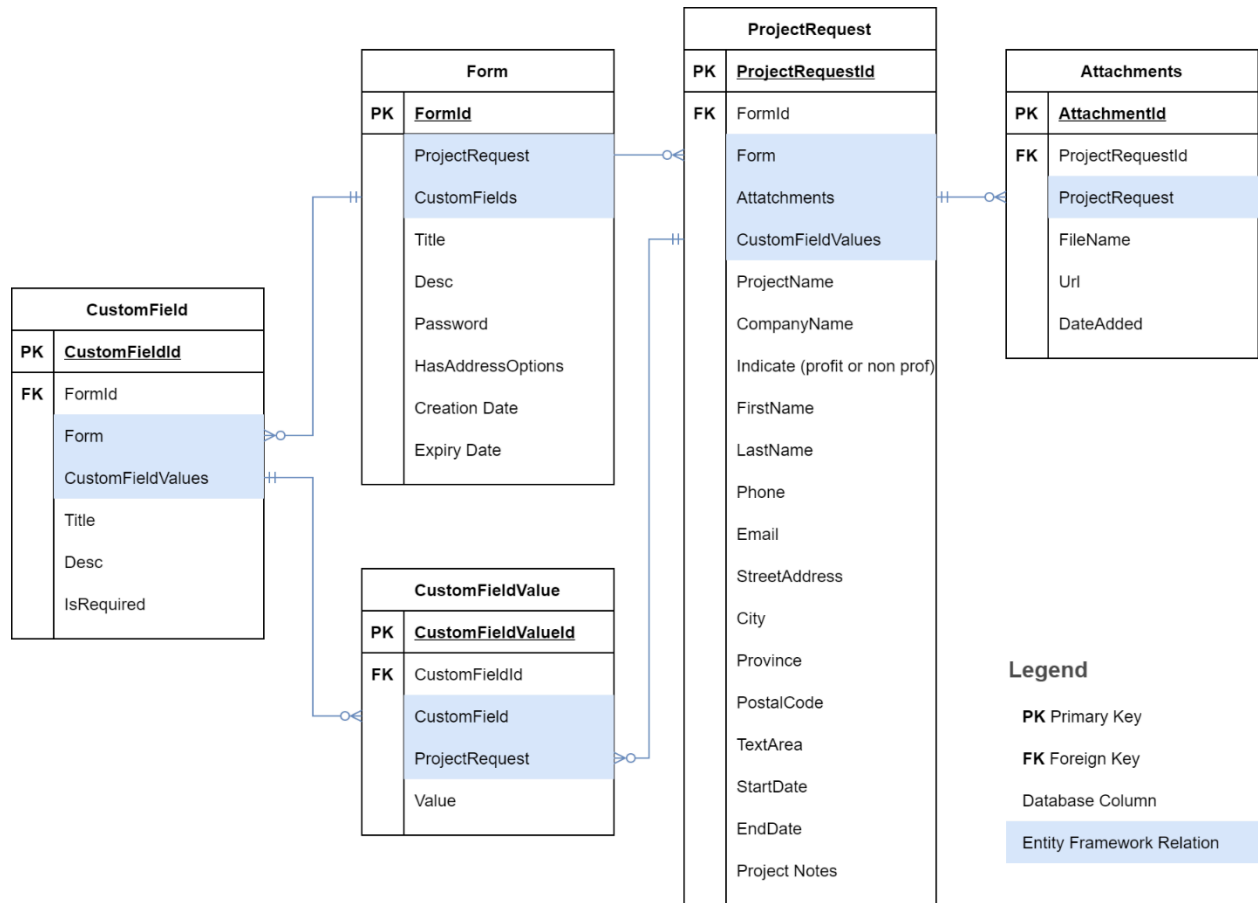


FIGURE 7 - DATABASE STRUCTURE DIAGRAM

4. TESTING, RESULTS, AND ANALYSIS

4.1 DEVELOPER TESTING

Developers have manually tested the application prototype during its development to ensure security, ease of use, and to leave as little bugs as possible for the next team developing the application. During future development and refactoring of the code, it is recommended that the next team implement unit testing to automate the testing process. This way, complicated testing scenarios such as file uploading and submitting complex project requests will take less time and can be standardized to ensure that any small change does not create bugs in the application.

4.2 CLIENT TESTING

The client can test the application through the web application server currently managed by dream development on the Azure cloud platform. By simply going to the link provided to the client exclusively, the client can see the project as it's developing and report bugs or suggest new features. After testing the current prototype on Dream Developments live servers, the client has suggested some changes that will be outlined for the next team to develop.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

Our team has succeeded in providing all the required functionality of the application. In the middle of developing the application, it was decided that the project will be passed to another team who will develop the project even further than we originally envisioned. So, while we have met all the requirements asked for in the final product, there is still more work to be done. Dream development has created a list of future recommendations from the client, required recommendations from our team, and suggested recommendations for the next team to further develop the application.

5.1.1 CLIENT RECOMMENDATIONS

After thoroughly testing the prototype, the client has suggested the addition of customizable fields that are visible only to the dashboard users and apply to every project request. This recommendation should be confirmed with the client as it may change by the time the next team starts development.

5.1.2 SECURITY REQUIREMENTS

Due to the team's focus on the functionality of the application, two additional security requirements are not yet implemented. The application must have features to prevent two types of threats to the client and application itself. One is a system to scan attachment files for viruses, and the other is a system to prevent bots from spamming forms or brute forcing the login password. Both factors should be when refactoring and adding new features to the application.

5.1.3 ADDITIONAL RECOMMENDATIONS

Other additions are suggested by our team are based on what we have learned developing this application. They are not necessary but may aid in future development

- API style refactor of attachments to simplify the calls to the database and blob storage.
- Add different data types to custom fields such as dropdowns or checkboxes.
- Automated unit testing back-end front end for complex functions such as file attachment and project submission.

6. REFERENCES

- [1] Microsoft, "Microsoft identity platform documentation," Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/develop/>. [Accessed 10 December 2021].
- [2] S. Smith, "Overview of ASP.NET Core MVC," Microsoft, 10 February 2021. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-6.0>. [Accessed 10 December 2021].