**Knox County Foundation - Online Vocational Scholarship**

**Requirements Document**

**Preface:**

The purpose of this document is to provide a comprehensive description of the online scholarship application system. This is the first version of the document. This version outlines the functionality and constraints of the designed system. This document is intended for both stakeholders and system developers. The content of this document is a subject to change as new user and system requirements are discovered.

**Introduction:**

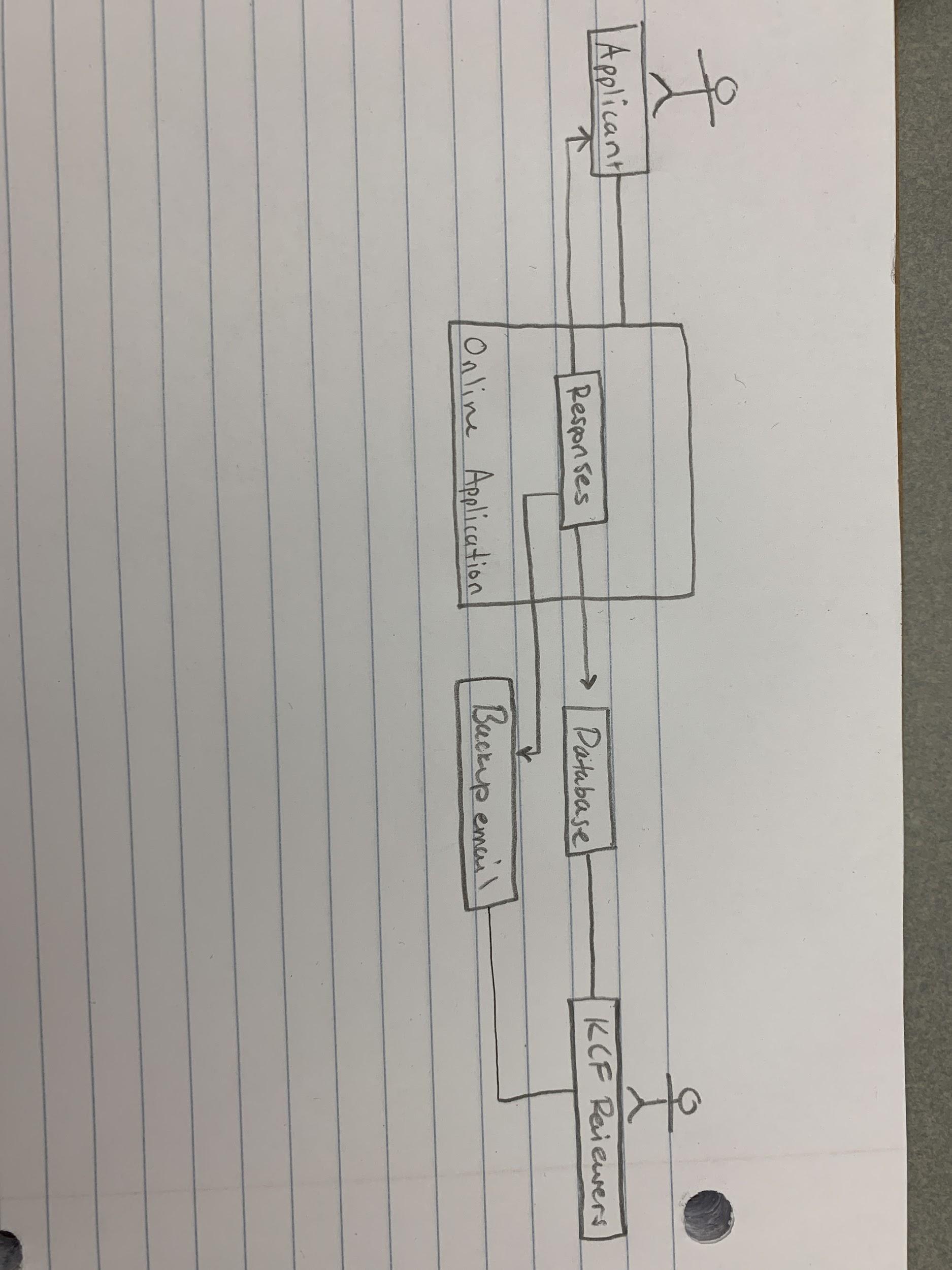
This system will allow the vocational scholarship application offered by the Knox County Foundation to be submitted online by applicants. Applicants will go on the KCF website, create an account to ensure that the applicant is human, and fill out the form online. They will be able to attach files as necessary (FAFSA, transcripts, financial aid packages). Once they submit their responses, the data from their application will be sent to the database containing all of the applications. The applicant will be sent a copy of their responses, and a copy will also be sent to KCF as a backup in case the database fails. Reviewers at KCF will be able to view a list of all the applicants and click on the individual applicants to review their answers. Reviewing the applications and seeing the database will be restricted to certain users as it holds sensitive data.

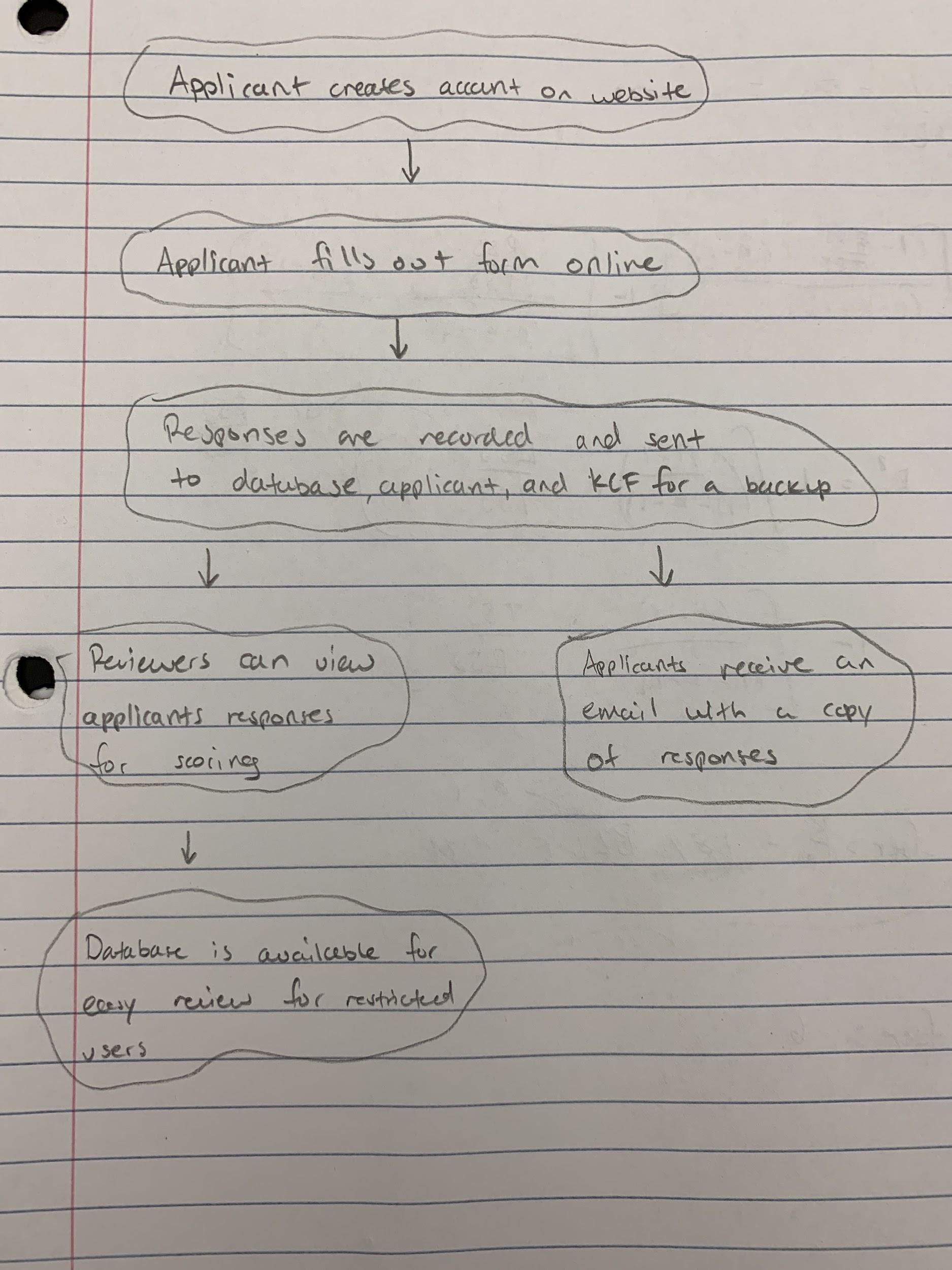
Applicants will be able to submit only ONE application to the vocational scholarship per year. After submitting an application they will be able to log-in and see the status of their application.

**Glossary:**

|  |  |
| --- | --- |
| Term | Definition |
| Central database | Primary database that stores information of applicants |
| Reviewer | Employees at the KCF that have access to the database and application responses for scoring and marking |
| KCF | Knox County Foundation |
| Primary key | The distinguishing variable between entries in a database. In our system it will be the email so each application will be distinguished from another by the user email. |
| Functional requirements | Services that are provided by the system. They describe how the system behaves under certain scenarios. |
| Non-functional requirements | Constraints on the functionality of the system including those relating to speed and performance of the system, and confidentiality issues. |
| User requirements | List of actions users are able to do with the system. |
| System requirements | List of features and functions of the system. |
| Backend database | Database accessible to authorised personnel of Knox County Foundation |
| Use case | List of actions that users can perform when using the system |
| Pre-condition | Condition that must be met prior to performing a system function |
| Trigger | Action required of a user to start an interaction with the system |
| Post-condition | The result of performing a system function |
| Exception path | It is an action taken in case of a fault condition |

**System model:**





**User requirements definition:**

Functional requirements:

**Applicant use case:**

Create account

* Brief description: The applicant creates an online account on the Knox County Foundation website.
* Step by step description:

1. Applicant open a webpage to create an account
2. Applicant fill in full name and an Email address
3. Applicant receives an email containing a link that sends them directly to the application

Fill in and submit application

* Brief description: The applicant opens a new application page online, then fills in and submits the application online
* Step by step description:

Before this use case can be initiated, applicant must have a valid account

1. Applicant opens a new application page
2. Applicant enters personal information details
3. Applicant enters responses to application questions
4. Applicant chooses “Submit” button to submit application
5. System validates input for correct format

**Reviewer use case:**

Reviewing submitted application

* Brief description: The reviewer logs in to the system, opens and reads a submitted application
* Step by step description:

Before this use case can be initiated, applications must have been submitted

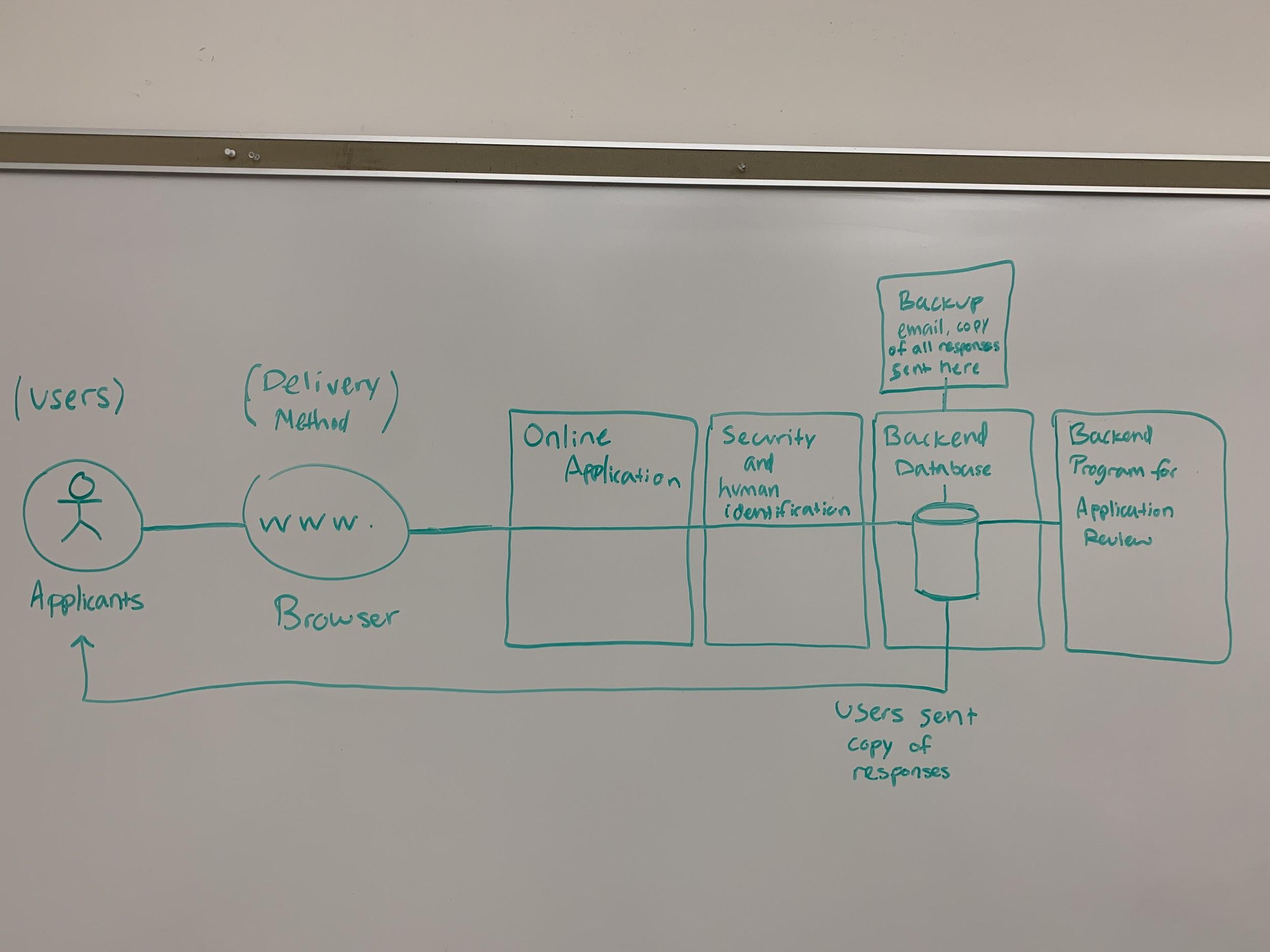
1. The authorized reviewer logs in to the system database of applications
2. The system checks to ensure that the user is authorized to view application data
3. If authorized - reviewer is able to open and review any submitted applications from the database
4. If not authorized - gets a pop-up message “Access not allowed”

Non-functional user requirements:

1. User accounts must be secure to protect sensitive data, with Email verification to prevent robots submitting fake applications.
2. Program must be finished before the end of the semester, first finish the online application then work on the central database.
3. The confidential information of the applicants has to be securely stored and be accessible exclusively to authorized personnel of the Knox County Foundation.
4. System component responsible for loading applicants information into the central database would be implemented in C++. Application page will be written in HTML but coded with javascript that will communicate with the database in C++.

**System architecture:**

In this diagram, the overall system architecture is shown. The lines depict online connectivity between each of the modules. Applicants visit the online application via a browser where the system then checks for human authenticity. This is to avoid fake submissions of the application. Security measures are high here because applicants will be submitting sensitive information. Once they submit their responses, their answers are saved in the backend database. A copy of their responses will also be emailed to them as well as to a backup email incase the database ever fails. Reviewers will then be able to view all of the submissions and judge accordingly.



**System requirements specification:**

**Create account:**

|  |  |
| --- | --- |
| **Use Case Name** | Create account |
| **Trigger** | The user selects create account on Vocational scholarship webpage |
| **Precondition** | Applicant has a valid Email address that could be accessed |
| **Basic Path** | 1. The webpage asks the applicant for a valid Email address and a password of at least 6 characters of numbers and letters. 2. The applicant clicks create account, and a confirmation Email is sent to the applicant with a link to verify the account 3. Applicant clicks on the link and is direct back to the application page with the account created 4. The link directs user to the online application form with the account successfully created 5. Once the account is created, display the online application form to the applicant |
| **Alternative Paths** | If in step 1 the password or Email address is invalid, ask for the user to fill in again |
| **Postcondition** | An account is created where the applicant can sign in with an email address and password |
| **Exception Paths** | The applicant can stop creating the account at anytime |
| **Other** | None |

**Fill in and submit application:**

|  |  |
| --- | --- |
| **Use Case Name** | Fill in and submit application |
| **Trigger** | The user types in their responses to all of the fields on the form, attaches the necessary files, and presses submit once finished |
| **Precondition** | They have created an account on the webpage |
| **Basic Path** | 1. The application has spaces for the applicant to fill in information and type in responses 2. The applicant enters responses and click submit 3. If application reponses meet all requirements, it is sent and saved to the database |
| **Alternative Paths** | If the application responses are not in the correct format, applicants would be notified via pop up prompting them to fix the responses.  Once the application is successfully submitted, a copy of the user’s responses are sent to their email as well as a backup email in case the database fails. |
| **Postcondition** | Their answers are securely kept in the backend database. |
| **Exception Paths** | The application can be closed at any time however their responses will not be saved. |
| **Other** | None |

**Open submitted application:**

|  |  |
| --- | --- |
| **Use Case Name** | Open submitted application |
| **Trigger** | The reviewer selects a completed application in the database |
| **Precondition** | There must be completed applications in the database |
| **Basic Path** | 1. Authorized reviewer logs in to the system database 2. System checks that reviewer is authorized 3. If authorized, All completed applications stored in the database and displayed to the reviewer 4. The reviewer clicks on the application to be reviewed |
| **Alternative Paths** | If reviewer is not authorized, error message will appear |
| **Postcondition** | The application is reviewed by the reviewer |
| **Exception Paths** | The reviewer may close the application anytime |
| **Other** | None |

**System Evolution:**

The first few versions of the system will be simple. It will allow the reviewers to see the database and applications individually. The system will be built such that if the KCF has the need to add additional functionality to the database and application viewing system, they should be able to easily. There will not be any hardware evolution changes as the system resides entirely online.

**Appendices:**

Database requirements - the database will be organized by the questions on the application. The primary key will be their email address.

Personal details data: name, address, city, state, zip, phone, email, date of birth, gender, last 4 digits of SSN

Resume type data: previous education which includes type (hs, college, etc), name, degree attained, and completion date

Current education which includes college/ school name, admission status(applied, accepted, enrolled), anticipated program, cost of program, anticipated start date, first payment due date

Personal goals which include 4 essay type questions about the applicant

Data being attached: High School transcripts, FAFSA, financial aid packages

Final data points: Whether they agree to a certification of truth statement(yes or no), applicant signature, date, and parent or guardian signature (if under 18) with date