**System Design**

**Initial High-Level Design**

Our initial high-level design used in developing the application is divided into three components:

1. Experimenter User Interface Web Forms
2. Subject User Interface Web Forms
3. Business Layer and Storage Libraries

The steps for the flow of data through the application are as follows:

1. Read Amazon Web Services S3 file structure and populate Experimenter Form drop down lists.
2. Input to the Experimenter UI to pick task type, content, duration and feedback type for each Subject task.
3. Submit task list from Experimenter UI to Subject UI.
4. Take more input from the Test Subject in the Subject UI as form answers.
5. Output these answers to responses.csv.
6. Store responses.csv to Amazon S3 Storage Services to appropriate folder.

The round-trip engineering components consisted of the following technologies:

1. ASP.NET version 4.6.1 with C# 7.0 as a web development framework.
2. Azure DevOps and BitBucket Cloud used as source version control platforms.
3. Third Party Libraries as helper methods for audio\video playback and timer functions.
4. Amazon S3 Software Development Kit as a collection of methods for cloud storage functions.
5. Stylus input tracking library as a list of functions to gather data about the user input.
6. Visual Studio 2017 plus AWS toolkit is as an Integrated Development Environment.

**Final High-Level Design**

The final high-level design is the same as the initial design except that an additional function to pass data from the experimenter module to the subject module has been added and is used to bridge the invocation and data passing between two ASP.NET project types. This was done to ease the integration of the two modules within the application.

This section could include various UML diagrams including, but not limited to *Use Case diagrams* or *Sequence diagrams*. Include only if it helps better understand the project.

Implementation Details

Implementation consisted of producing and reviewing the wire frames for the user interfaces and developing the business layer and storage library methods to be called from the user interface code behind files. ASP.NET produced the correct presentation layer and java script to be rendered by the browser seamlessly making this solution compatible with any hardware capable of running a so called “uplevel” web browser, that is a browser that supports at least the following:

* ECMAScript (JScript, JavaScript) version 1.2.
* HTML version 4.0.
* The Microsoft Document Object Model (MSDOM).
* Cascading style sheets (CSS).[1]

[1] <https://docs.microsoft.com/en-us/previous-versions/aspnet/x3k2ssx2(v=vs.100)>

An interesting aspect of the implementation was how to make the timer common to the subject forms and expire each task at the right time. This technical challenge was accomplished by:<Ramanjit fills in detail>. Another challenge was to ….

**Testing Details**

The testing objective for the application is to ensure that either the experimenter or the test subject has an intuitive and error free user experience using a variety of hardware and web browsers.

The following web browsers were tested with the application:

1. Google Chrome version 70.0.3538.77
2. Internet Explorer 11 to Microsoft Edge 42.17134.1.0
3. Safari 11.1.2 to 12.0.1

The following tests were performed on each browser to verify the application:

1. Use the same three versions of type task mix and test subject responses on each browser. Does the application produce equivalent outputs?
2. Is the application responsive for each mix of tasks and responses?
3. Is the application intuitive or is the content distracting to either the user or experimenter?
4. Does the application guide or prompt the user regarding proper input formats?

**Writing prompt for the above section is:**

* *System details: In this section you must provide the high-level design, implementation and testing details.* 
  + *Show and explain the high-level system architecture with various important blocks in the system. In many projects, the initial design and the final design differ somewhat. If the differences are interesting, write about them, and why the changes were made. If your design was not implemented fully, describe which parts you did implement, and which you didn't.*
  + *Discuss the most important/interesting aspects of implementation. It probably won't be possible to discuss everything. Give a rationale for what you do discuss.*
  + *Explain how the program system was verified.*