1. **Introduction:**

“Window Software for Human Studies” project is used to build an interface in C# that will dynamically deliver the contents to students and marking the beginning and end times. This is the universal website that can run on the computer, Microsoft surface or writing tablet. The student has to response with the stylus. The software will be used to capture all the relevant stylus variable (trajectory, speed etc.). The responses from subjects are saved in the AWS S3 bucket.

This application has 2 modules namely, Subject Interface and Experimenter Interfaces integrated through Amazon Web Services (AWS). Experimenter module is responsible for creating exams and store it in AWS database. Subject module would be responsible for taking exams created by Experimenter. Responses of the exams taken by Subjects would be stored in AWS database and then in central repository called Subject Book. Exams created by Experimenter would also be stored in Subject Book.

* 1. **Background:**

“Window Software for Human Studies” is an application for getting information about users, also called Subjects, through exams. Subjects will be asked to take exams and the responses from users will be saved in AWS database and then in Subject Book, which will eventually be used in research for managing stress of humans. This application is a part of ongoing research project being conducted in Computational Physiology Lab (CPL). It was founded in 2002 by Dr. Ioannis Pavlidis and since it has been appreciated worldwide for its active involvement in research activities. At this moment the lab has three research lines and an educational research effort on science ethics. CPL is an interdisciplinary lab and draws additional expertise from partner labs in The Mayo Clinic, The University of Texas Medical School, and The Houston Methodist Hospital.

There are three software modules that are being used by this lab, namely: Subject Book, U- Interface, and S-Interface for management of stress. These interfaces communicate via Google drive. The S-Interface has been built as a modular system in C#. It is a communal software development project, where anybody can contribute modules with specific requirements. This interface has real time multiple sensor channels during the experimental session, uploading the collected the data to a designated Google Drive. The U-Interface captures the subject’s response to questionnaires’, uploading the collected data to a designated Google Drive. Once in the Google Drive, much like the sensors data are curated and managed by Subject Book.

The “Window Software for Human Studies” is an application similar to U-Interface that will also contribute to the data about users stored in Subject Book through getting responses from users by asking them to take exams. Users, also referred as Subjects, will be asked to take exams consisting questions from various fields. These questions would be fed to Subjects by Experimenters. Experimenters are the one who are responsible for creating exams for Subjects.

* 1. **Purpose:**

The purpose of the tool integrated with the project is to collect data about Subjects. This data will be helpful to analyze users. The project will develop two User Interfaces (UI) that are integrated with the AWS. These interfaces are:

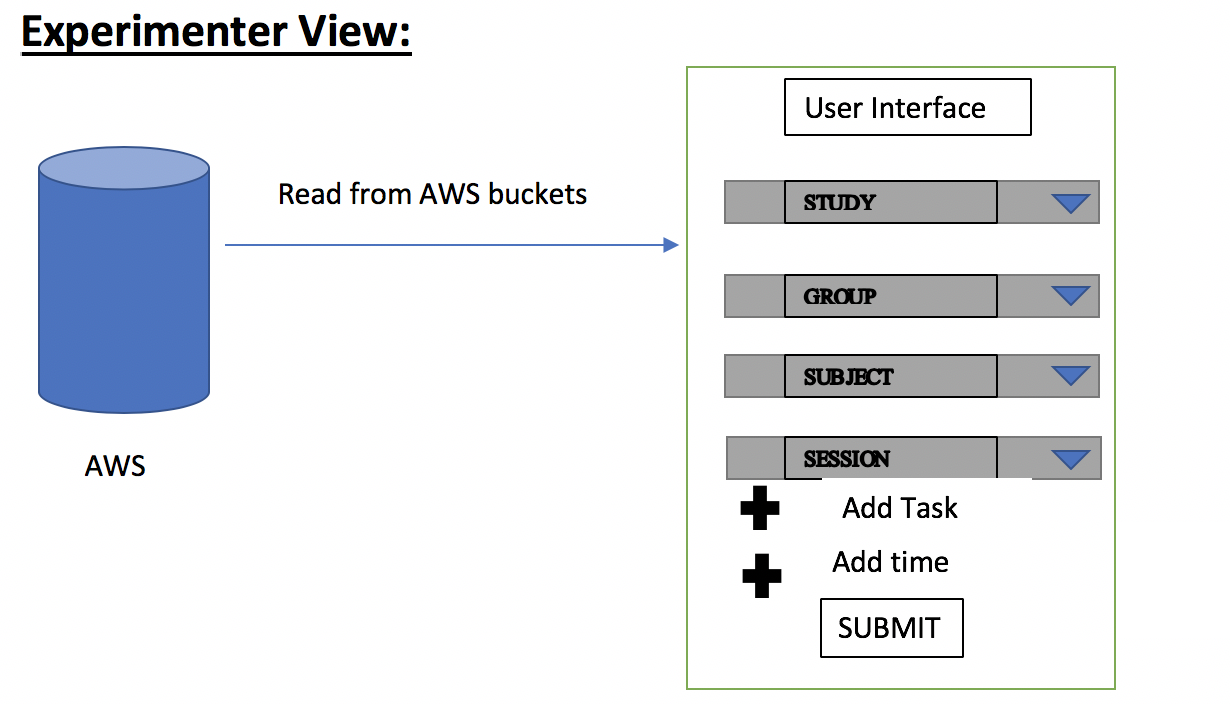
1. Experimenter Interface
2. Subject Interface

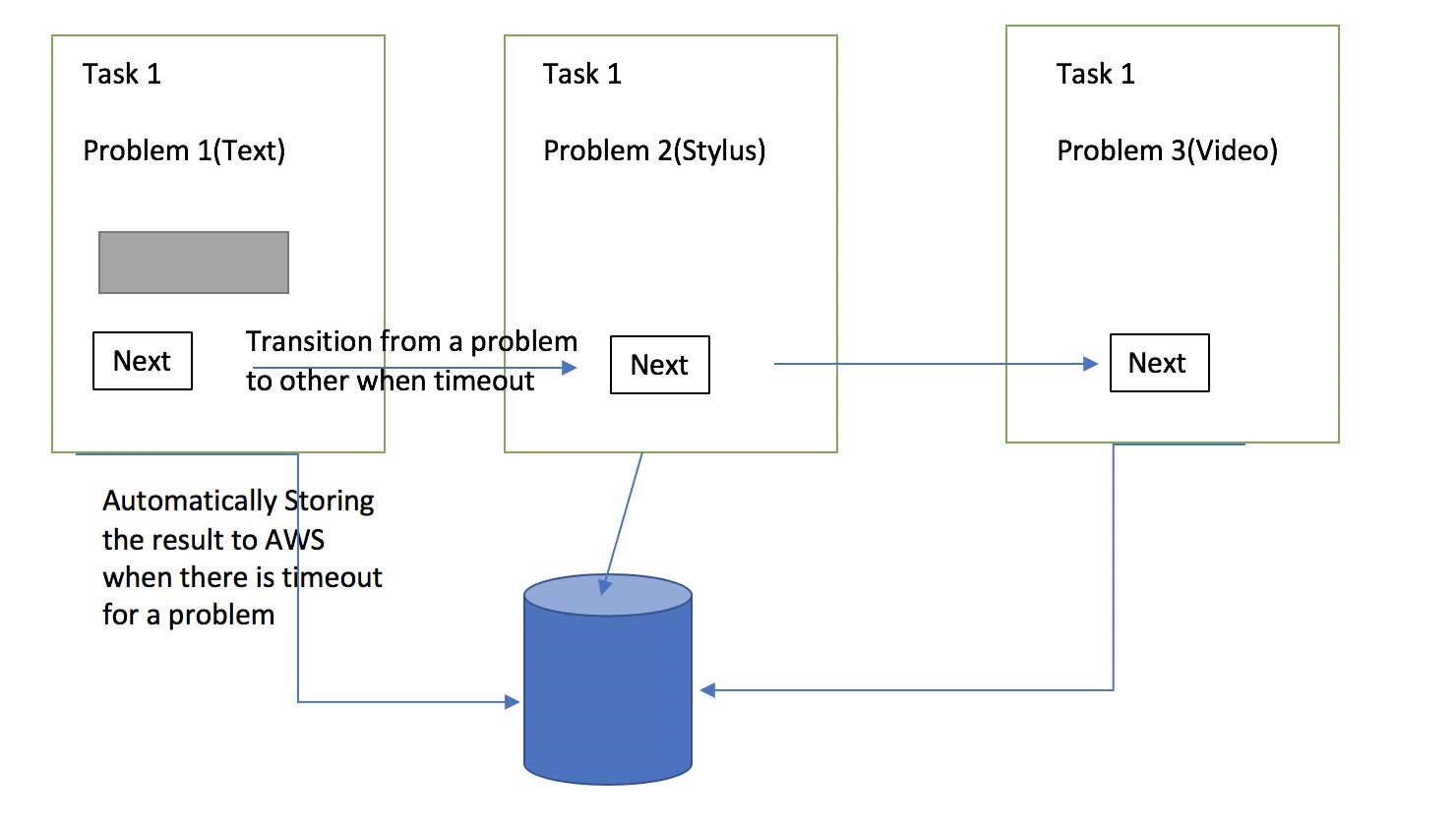
**Experimenter Interface**: This interface is used to create tasks for subjects. Each task contains questionnaire of various formats like audio, video, text and image. Questions will be timed and untimed as well. Experimenter can set the duration of the task. Experimenter can load the questions that are predefined in the AWS or can make the new task. For loading a new task, experimenter has to set the study, group, subject and the session. These Variable are saved in the config file in the AWS. For creating task, experimenter needs to define the task type as well as the duration of the task.

**Subject Interface**: Subject interface is designed for the subjects to do the task. These tasks are timed or untimed. For the timed questions, subjects are required to answer with the allotted time showing on screen. If the user fails to provide the response, user will be navigated to other view that contain the other problem and blank response will be saved. Subject won’t get the option to go back to answer that question. These responses are capture and stored in the AWS S3 bucket. Each task response as well as task duration is stored in the CSV file.

**AWS integration:** Experimenter is delivering the questions to subjects via AWS. Experimenter can send the predefined task to the users or can upload the new task for the user. For uploading the new task, session variable is used that dynamically deliver the questions to the subject interface. For sending, the predefined task, experimenter will interact with the AWS. For retrieving the task and uploading the responses, GetObject and PutObject functions of AWS are used.

The basic diagram of how this interaction works is given below:





* 1. **Project Scope:**

1. Stress is a big issue as it hampers productivity.
2. Measuring stress level is important in order to take actions to prevent it.
3. To prevent or to lessen the stress, gathering data is very important.
4. This application is used to gather required data for analysis purpose.