1. Introduction

The memory game application is designed for practicing facial recognition through memory stimulating exercises. The goal of the application is to improve the user’s ability to recall common names based on visual prompts, typically a photograph. The main consideration of design is to make the program as accessible as possible to users of the User role type. This is due to the anticipated variability in users’ prerequisite knowledge of computers and ability to interact with user interfaces. The design that was chosen to be implemented utilizes Microsoft Window’s Software Development Kit (SDK) with the .net framework. This is a freeware kit that allows developers to use the Window’s environment to create customized applications. Microsoft’s Visual Studio Integrated Development Environment, which may be downloaded for free, will be used to develop the game in the C# language, which is very similar to Java. This design was chosen as it is the most cost-effective option, while also providing all the necessary means for creating both a simple graphical interface and a voice recognition component. In general, two roles will exist in this program. These roles are discussed below, and Figure 1 describes the tasks available to each role, as they comprise the whole program.

1.1 Administrator Role

The default mode of memory application will give access to Administrator and User roles. Administrators will have open, create and update access to the user profile database. Administrators will be able to export and import scenario data and initialize a User role session.

1.2 User Role

Once an Administrator has initialized a scenario, the user interface is simplified to show only the items necessary to run the scenario. Users will use their voice or keyboard to input the name associated with the displayed picture at each round. (They will automatically be advanced to the next round upon completion). A score indicator will be provided after all rounds of the scenario have been run and the user interface will revert to reflect the administrator role.

Figure 1: User and Administrator cases
2. Subunits

2.1 System Overview

The application will consist of several subsystems. An intuitive user interface will allow the User to access the full functionality of the application through a menu system. Additionally a profile database will keep track of the Users of the program, their associated persons, scenarios, and photos. A handler class will be used to create files containing information on a scenario or load a scenario file. A speech handler class will be used to enable speech input for names during a User’s scenario session. Figure 2 below is a detailed break-down of all the classes, variables, and methods that will be necessary in the code.

![UML class diagram](image)

**Figure 2: UML class diagram**

2.2 User Interface (Administrator Role)

Upon entering the application, the User will be prompted to create or select a profile. The list of current profiles will be loaded from the profile database. Creating, editing or switching profiles can be accessed at anytime from the top menu bar of the program window. The default interface upon admittance will show a list of recent scenarios, a button to run those scenarios, and an option to create new scenarios. Creating or editing a scenario will bring the user to a list of persons attached to that user’s profile. Persons may be added to the scenario list in a one-to-many relationship so a person can be used more than once in a single scenario. Randomized scenarios can be created and run on the fly from this screen. Additional persons may be added from within the Administrator role user interface. A panel containing various fields pertaining to a person are available from the edit/create person menu option. A person and photo library will be maintained within each user profile database.
2.3 User Interface (User Role)

The User session is automatically initialized when a scenario is run. The application window then switches to full screen mode and the first round commences. A round consists of a photo being displayed to the User and the User speaking the correct phonemic pattern into the microphone or typing the correct text into the provided text box. The round is then scored for correctness and a new round is shown until all rounds in the scenario are exhausted. The user interface will then revert to the Administrator role and display statistics. Figure 3 below describes the sequence of events that would occur upon running the game for both the User and the Administrator roles.
2.4 Profile Database

The back end of the memory game application consists of a profile database. The profile database will be stored and encrypted in save files and handled through a Data Access Object. Figure 4 below describes this database in more detail.

![Database Diagram]

**Figure 4: Database diagram**

2.5 Scenario Save Handler

A specially designed class will be used to create encrypted scenario files. These files will consist of the person list associated with the scenario and any other settings needed to run the scenario. This will enable exporting and importing files between workstations or directories.

2.6 Speech Recognition

Speech recognition is an essential characteristic of the memory game application. The program will achieve this by using the System.Speech reference of the .net libraries. This reference will import Speechlib SDK 5.3 for Windows Vista operating systems and SDK 5.1 for Windows XP operating systems. A speech handler class will handle all speech interaction containing methods to start and stop listening, identify and compare phonemes, and store phonemes for comparison.

2.7 Required Hardware

A properly installed microphone is necessary for the speech components of the memory game application to work. Using the Windows speech training wizard increases the accuracy with which the speech libraries can recognize the User’s voice. The team anticipates the program will have minimal memory cost. Recommended system requirements will likely be 512MB of memory on XP machines and 1GB of memory on Vista machines.
2.8 System Dependencies
The memory game application requires Windows XP or Windows Vista. A CD-Rom drive is required for installation. .Net 3.0 libraries must be installed on the system for the application to run voice components properly.

2.9 Testing
In general, each of the components discussed will be tested class by class, method by method. The majority of the development of the game will most likely be spent debugging. The speech component will probably require the most testing, simply by writing code to accept certain names and then speaking the names into the microphone to test if they are recognized in the context of the game. Testing with multiple user voices will also be an important component to ensure that the speech component will work well with more than one user.

3. Realistic Constraints
One of the constraints that will be encountered by the memory game application is that the game will be restricted to computers running on Windows Vista or XP. A Mac operating system would require another version of the software, which could be supplied if the client expresses interest in this.

Another problem is the accessibility of the software game. The game is being designed to be as user friendly as possible, but since this software’s purpose is aimed at patients with cognitive impairments, especially those with Alzheimer’s and dementia, most of them will not be able to install and run the software without some assistance.

In terms of manufacturability, as this project requires the design of software, there isn’t anything to be physically created, except for the CD-Rom, containing the data to be installed on computers to run the game. Depending on the type of operating system being used, the creation of the installation software may differ. The microphone to be used with the game will be purchased. Since there are no other costs required for this project, except the purchase of some low-cost CD-Rom discs, the team believes it is feasible to purchase a high end microphone which makes use of filters and noise-cancelling technology to maximize the game’s ability to pick up sounds and use them in the context of the game.

4. Safety Issues
There are no immediate safety issues that this software package will present. The only piece of physical hardware that is implemented in the design is a microphone unit which will be carefully chosen for its ease of use and quality. Because there will be staff present to help the designated clients perform this game, safety does not play much of an issue.

5. Impact of Engineering Solutions
This design of a memory game application to enhance memory recall in people with Alzheimer’s or other short-term memory impairments will have a societal impact on those who make use of the software. A human being’s ability to interact with those around him/her is an irreplaceable aspect to life, which, if impaired by cognitive disabilities, can have heart-breaking consequences for those affected by the disability and their loved ones. Being able to recognize the faces of people they love, say their names, and engage them on a level similar to that
before their memory impairment began can greatly increase quality of life for those with memory impairments. On a larger scale, this software design is a small addition to the efforts being made to understand memory loss induced by advanced age and especially by Alzheimer’s Disease and dementia. It uses technology to address the issue of poor memory recall and to improve it through repetitive viewing of images. Hopefully its use will inspire patients, family, friends, and caretakers to believe that there is hope for people with these impairments, that more research is needed to help battle these losses, and that technology can play a helpful role in the future.

6. Life-Long Learning

During the course of designing this project, new skills have been learned, and old skills have been refined. Traits such as time management, responsibility, teamwork, and leadership have also played a large role during the evolution of the designs. Each one of these traits is vital to working in a real world environment, and by practicing such traits Team 8 has gained a valuable foothold for future endeavors.

Time management played a large role because in order to create a software application a specific timeline needed to be laid out in order to get certain aspects of the project done on time so that future work could be built upon its foundation. Time management is an important skill to learn and refine because it is applied in all kinds of jobs and different situations. Also team work and responsibility are traits which are desired when working in the real world because, in most applications, one individual is not responsible for a project, but a group of individuals is. By learning how to work in a team atmosphere and having the responsibility to get the designated work done, you can apply these traits in a favorable way to future opportunities.

The design of this software game helped to greatly improve the programming skills of the team. Being given the opportunity to become familiar with Windows SDK and the .net framework is an excellent opportunity to add to the team’s computer knowledge, which is very useful in a professional engineering environment.

When researching the way in which this design software package would be implemented, a lot of research needed to be performed and an assessment had to be made about the best way in which to approach the project. These in itself provided valuable experience to the team because being able to research and learn how a system works, and then determining the best way to go about completing the task, is a very practical skill that can be used throughout life.

Overall, the skill sets which were learned and applied during the design process of this project provided valuable experience to the team. By refining such traits as leadership, teamwork, and time management the team was able to work well and stay on task in order to complete the design phase for the project. Also a practical application of learning how a system works and applying it to solve a problem was learned. These skills provide valuable experience for implementing what was learned in the classroom into everyday life, and help provide a base which can be built upon for future careers.