**Main shortcomings**

**Missing Architecture diagram,**

**Detailed design incomplete – data and Procedure inside each object/module is missing**

**Standardized Application System (SAS)**

**Software Design Description (SDD)**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Revision Description | Pages Affected |
| 1.0 | 8-04-07 | Initial Delivery | N/A |

**Table of Contents**

[1. Introduction 4](#_Toc174017817)

[***1.1.*** ***Purpose*** 4](#_Toc174017818)

[***1.2.*** ***Scope*** 4](#_Toc174017819)

[***1.3.*** ***Definitions*** 4](#_Toc174017820)

[2. References 4](#_Toc174017821)

[3. Decomposition Description 4](#_Toc174017822)

[***3.1.*** ***Module Decomposition*** 5](#_Toc174017823)

[**3.1.1.** **Authentication module** 5](#_Toc174017824)

[**3.1.2.** **Application Entry Module** 5](#_Toc174017825)

[**3.1.3.** **Database Controller Module** 5](#_Toc174017826)

[**3.1.4.** **User Control Module** 5](#_Toc174017827)

[**3.1.5.** **Notification/Reminder module** 5](#_Toc174017828)

[**3.1.6.** **Evaluation Controller Module** 5](#_Toc174017829)

[**3.1.7.** **Statistics Module** 5](#_Toc174017830)

[**3.1.8.** **Status Module** 5](#_Toc174017831)

[**3.1.9.** **Student admissions interface module** 5](#_Toc174017832)

[**3.1.10.** **University Faculty/Staff information interface module** 5](#_Toc174017833)

[**3.1.11.** **Student information interface module** 5](#_Toc174017834)

[**3.1.12.** **Mail forwarding interface module** 5](#_Toc174017835)

[**3.1.13.** **Batch process module** 5](#_Toc174017836)

[**3.1.14.** **Administration module** 6](#_Toc174017837)

[***3.2.*** ***Concurrent Process Decomposition*** 6](#_Toc174017838)

[***3.3.*** ***Data Decomposition*** 6](#_Toc174017839)

[**3.3.1.** **Undergraduate Application table** 6](#_Toc174017840)

[**3.3.2.** **Graduate Application table** 6](#_Toc174017841)

[**3.3.3.** **Undergraduate Relation** 6](#_Toc174017842)

[**3.3.4.** **Graduate Relation** 6](#_Toc174017843)

[**3.3.5.** **Users** 6](#_Toc174017844)

[**3.3.6.** **Statistics** 7](#_Toc174017845)

[4. Dependency Description 7](#_Toc174017846)

[***4.1.*** ***Intermodule Dependencies*** 7](#_Toc174017847)

[***4.2.*** ***Interprocess Dependencies*** 7](#_Toc174017848)

[***4.3.*** ***Data Dependencies*** 8](#_Toc174017849)

[5. Interface Description 9](#_Toc174017850)

[***5.1.*** ***Module Interface*** 9](#_Toc174017851)

[**5.1.1.** **The Model Subsystem** 9](#_Toc174017852)

[6. Detailed Design 11](#_Toc174017853)

[***6.1.*** ***Application Submission Subsystem Module detail*** 11](#_Toc174017854)

[**6.1.1.** **Application submission subsystem** 11](#_Toc174017855)

[***6.2.*** ***Data Detailed Design*** 14](#_Toc174017856)

[**6.2.1.** **Users** 14](#_Toc174017857)

[**6.2.2.** **Statistics** 15](#_Toc174017858)

[**6.2.3.** **Undergraduate Application Table** 15](#_Toc174017859)

[**6.2.4.** **Graduate Application Table** 17](#_Toc174017860)

[**6.2.5.** **Graduate Relation** 18](#_Toc174017861)

[**6.2.6.** **Undergraduate Relation** 18](#_Toc174017862)

**List of Figures**

[Figure 1 Table Relationships 8](#_Toc174017863)

[Figure 2 SAS Submission Package 9](#_Toc174017864)

[Figure 3 Applicant Entry Screen 12](#_Toc174017865)

[Figure 4 Login Screen 13](#_Toc174017866)

[Figure 5 Applicant Status Screen 14](#_Toc174017867)

1. **Introduction**
   1. ***Purpose***

The purpose of this document is to describe the detailed design of the Standardized Application System (SAS). It will show the client the design and details of SAS and show the developers how SAS should be implemented.

* 1. ***Scope***

The scope of this document describes the following about SAS:

* Its Decomposition
* Its Dependencies
* Its Interface Descriptions
* Its Detailed Design

This document focuses on the details of SAS and does not describe the details of the systems that SAS interfaces with.

* 1. ***Definitions***

|  |  |
| --- | --- |
| Chair of the Committee | Responsible for final approval of graduate applicants. |
| Department Staff Advisor | Responsible for evaluating undergraduate candidates. |
| Graduate Program Committee | A committee of department faculty responsible for evaluating graduate candidates. |
| Undergraduate Program Committee Chair | Responsible for final approval of undergraduate applicants. |

1. **References**

The following documents were used to develop this SRS

* Summer 2007 Term Project Requirements-Dr. Mansour Zand, June 2007
* SyRS for Standardized Application System (SAS), prepared for Dr. Mansour Zand, July 2007.
* SRS for Standardized Application System (SAS), prepared for Dr. Mansour, July 19, 2007
* SPMP for Standardized Application System (SAS) prepared for Dr. Mansour, July 26, 2007
* Sample SDD for Ophelia’s Reservation System, prepared for Dr. Mansour Zand, April 19, 2005
* IEEE Std 1016, 1998 Edition – Software Design Descriptions

1. **Decomposition Description**
   1. ***Module Decomposition***
      1. **Authentication module**

This module supports login for the users of SAS as well as allows for a new applicant to register and receive a user name and password.

* + 1. **Application Entry Module**

This module allows an application to be entered into SAS either by the applicant or a registrar clerk.

* + 1. **Database Controller Module**

This module controls input and output to the SAS database.

* + 1. **User Control Module**

This module controls user activity and user accounts.

* + 1. **Notification/Reminder module**

This module performs the notification and reminders for SAS. It is responsible for the automatic generation and sending of notifications and reminders to the appropriate parties.

* + 1. **Evaluation Controller Module**

This module controls the application evaluation process and the flow of screens used to evaluate whether an applicant is accepted or rejected.

* + 1. **Statistics Module**

This module allows the applications’ statistics to be viewed

* + 1. **Status Module**

This module allows an application’s status to be checked and/or updated.

* + 1. **Student admissions interface module**

This module controls interfacing with the Admissions System

* + 1. **University Faculty/Staff information interface module**

This module controls interfacing with the Faculty/Staff information System

* + 1. **Student information interface module**

This module controls interfacing with the student information system

* + 1. **Mail forwarding interface module**

This module controls interfacing with the mail distributor system

* + 1. **Batch process module**

This module automatically starts the notification module to check for notifications/Reminders that need to be sent out.

* + 1. **Administration module**

This module controls the administration activities

* 1. ***Concurrent Process Decomposition***

SAS must be able to handle multiple users concurrently. This is necessary for the system to meet the requirements given. The only instance when this would be not so is during maintenance.

* 1. ***Data Decomposition***
     1. **Undergraduate Application table**
* This table keeps tracked of the undergraduate application data that the applicant has entered including the applicant’s contact information, high school information.
* It contains if the five criteria was meet. This is entered when the advisor reviews the application
* Advisor notes
* If the application was accepted/reject.
* If SAS is finished processing the application.
  + 1. **Graduate Application table**
* This table keeps tracked of the graduate application data that the applicant has entered including the applicant’s contact information, high school information, and College information
* It contains if the seven criteria was meet. This is entered when the GPC reviews the application
* GPC notes
* If the application was accepted/reject.
* If SAS is finished processing the application.
  + 1. **Undergraduate Relation**

This table keeps tracked of which users are working on which undergraduate applications

* + 1. **Graduate Relation**

This table keeps tracked of which users are working on which graduate applications

* + 1. **Users**

This table contains the basic contact information for the user, username, and password. It also contains the role that the user plays. The role gives the user access to his/her needed functions

* + 1. **Statistics**

This table contains the number of applications that are pending, accepted, or rejected by Graduate/Undergraduate

1. **Dependency Description**
   1. ***Intermodule Dependencies***

Almost all modules are dependant on other modules and in this case, this is acceptable as the system will still be very stable.

* 1. ***Interprocess Dependencies***

All execution threads run independent of each other. Although there is a sequential flow, it is implied more than structured that way.

* 1. ***Data Dependencies***



Figure 1 Table Relationships

1. **Interface Description**
   1. ***Module Interface***
      1. **The Model Subsystem**
         1. ***Application Submission Subsystem***

The application submission subsystem contains the applicants application along with the aggregate statistics and the attributes the applicant enters during the application process.

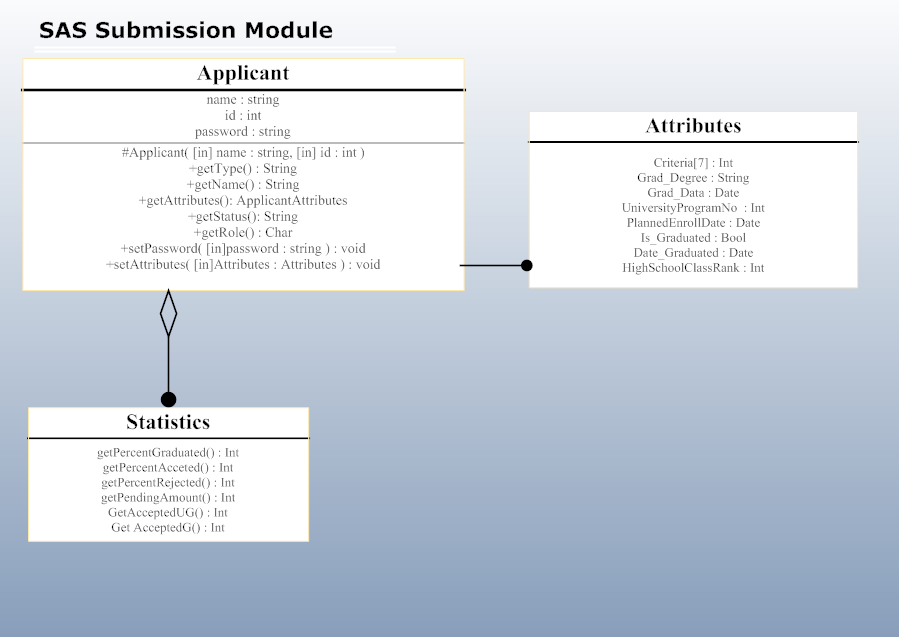


Figure 2 SAS Submission Package

* + - * 1. **Applicant**

The Applicant class provides interfaces for other packages to retrieve attributes from the specific applicant, along with name type and roles associated with the user. This applicant class is used in many modules across the SAS system for defining security, screens, and views inside of the system.

|  |  |
| --- | --- |
| **Applicant Methods and Properties** | |
| Name | Name of the applicant |
| ID | The User ID of the applicant, used internally to match up records by unique IDs |
| Password | The hash encrypted password for the applicant. This is hashed and is not human readable. |
| getType() | Returns the type of the applicant (Graduate or Undergraduate) |
| getAttributes() | Returns the attributes class for the applicant. This contains all of the criteria that the applicant has filled out as well as any notes that are on the application so far. |
| getStatus() | The current status of the application |
| getRole() | The role of the user in the system. Different users have different roles when they log in such as admin, applicant, OPC, etc… |
| setPassword() | Allows the user to set a new password after the initial password has been setup. |
| setAttributes() | Updates attributes for the applicant. |

* + - * 1. **Attributes**

Attributes provide an array of seven different criteria that the applicant will enter during the application process. Depending on if it is a graduate or undergraduate submission different fields will be filled out in this.

|  |  |
| --- | --- |
| **Attributes Properties** | |
| Criteria[] | An array of all the criteria that the applicant has filled out. For undergraduates this will contain 5 elements and for graduate students it will contain 7 elements |
| Grad\_Degree | The degree that the graduate has previously received |
| Grad\_Date | Day of the previous college graduate |
| UniversityProgramNo | The associated internal university number for the program that the applicant is attempting to enrolled in. |
| PlannedEnrollDate() | The date that the student plants on having for their first day of class. This will only be relevant if the student is accepted into the university. |
| Is\_Graduated() | Returns whether or not the student has graduated from high school before. |
| Date\_graduate() | Returned the date when the student has graduated from their high school. |
| HighSchoolClassRank() | The class rank of the student in high school. |

* + - * 1. **Statistics**

Statistics are cumulative collection of Applicants in the system. It returns various statistics on the percentage of applicants received, graduated, accepted, and returns the number of applications still pending, and finally the amount of undergraduate vs. graduate accepted.

|  |  |
| --- | --- |
| **Statistics Methods** | |
| getPercentGraduated() | Returns the amount (in percent) of applicants who have graduated from their highschool. |
| getPercentAccepted() | Returns the amount (in percent) of applicants who have already been accepted into the University. |
| getPercentRejected() | Returns the amount (in percent) of applicants who have been rejected into the University. |
| getPendingAmount() | Returns the amount of applicants who are still pending access into the University. |
| getAcceptedUG() | Number of accepted undergraduates into the University. |
| getAcceptedG() | Number of accepted Graduates into the University. |

1. **Detailed Design** 
   1. ***Application Submission Subsystem Module detail***
      1. **Application submission subsystem**
         1. ***Application Entry***
            1. **Grad/Undergrad Applicant Entry Screen**

Provides the form by which an applicant submits their application or a way for a registrar clerk to enter the data for undergrad applicants and grad applicants.

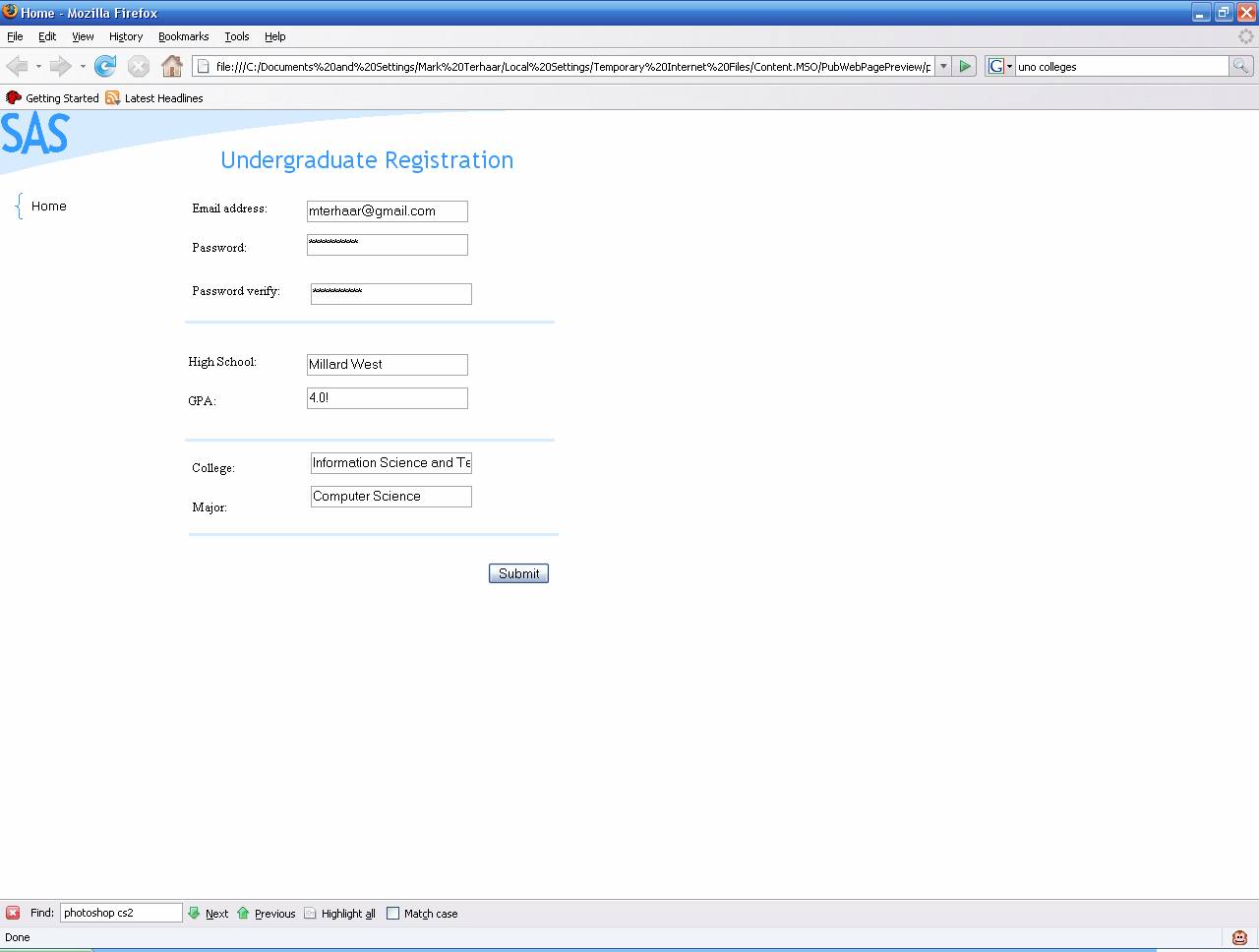


Figure 3 Applicant Entry Screen

* + - * 1. **Check Application Controller**

When invoked this object checks the application for errors before it is submitted to the database.

* + - * 1. **Submit Application Controller**

This controller is responsible for sending the application information to the database controller for storage and use by the rest of the system.

* + - 1. ***Users of this subsystem***
         1. **Applicant**

This is a type of user. This class includes undergraduate and graduate students.

* + - * 1. **Registrar Clerk**

This is the class of users that is responsible for entering mailed in applications into SAS.

* + - 1. ***Authentication Control***
         1. **Registration Screen**

The function of this screen is to allow a new applicant to acquire a username and password to access SAS.

* + - * 1. **Login Screen**

This is the screen by which users, in this case returning applicants, gain access to SAS and their specific tasks.

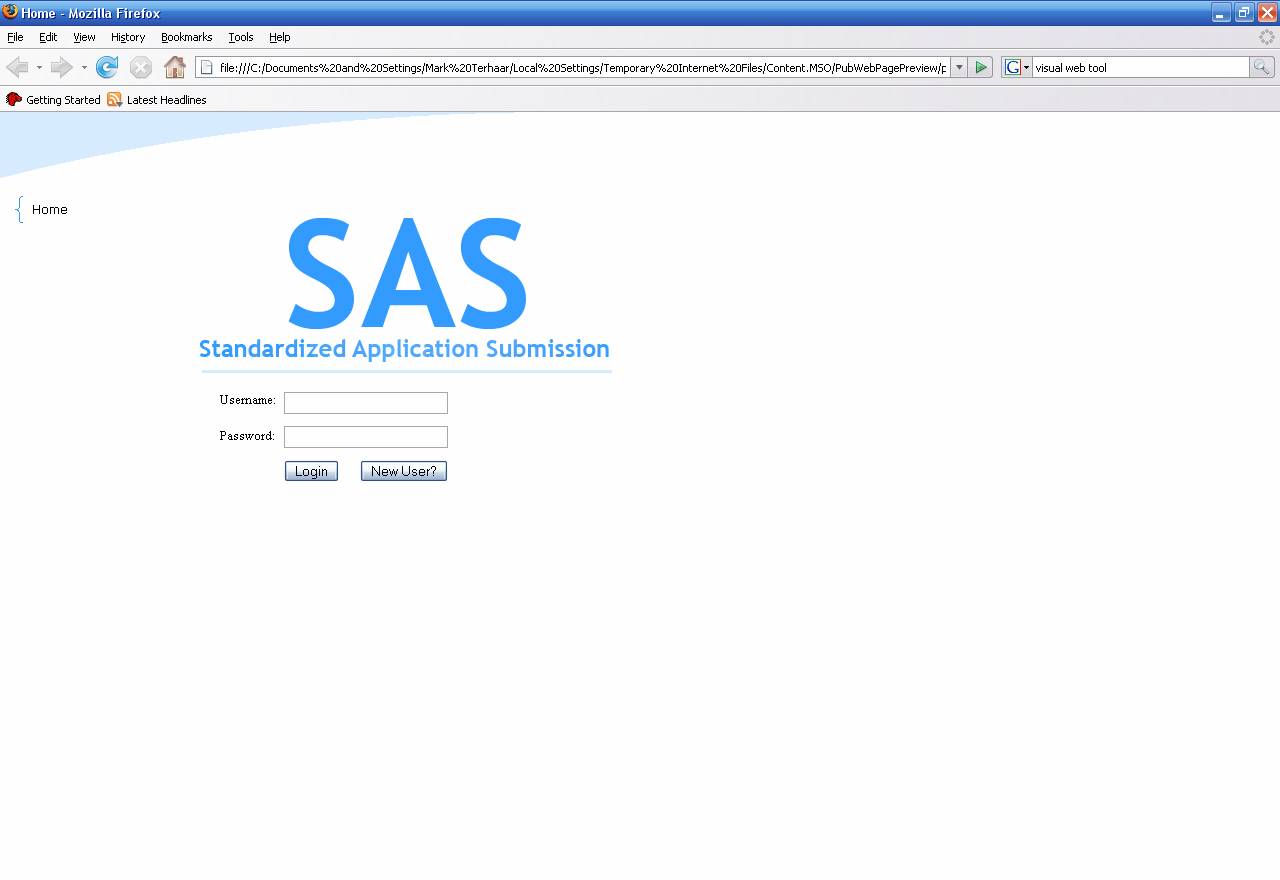


Figure 4 Login Screen

* + - 1. ***Status Module***
         1. **Applicant Status Screen**

This screen allows the applicant to check the status of his/her application.

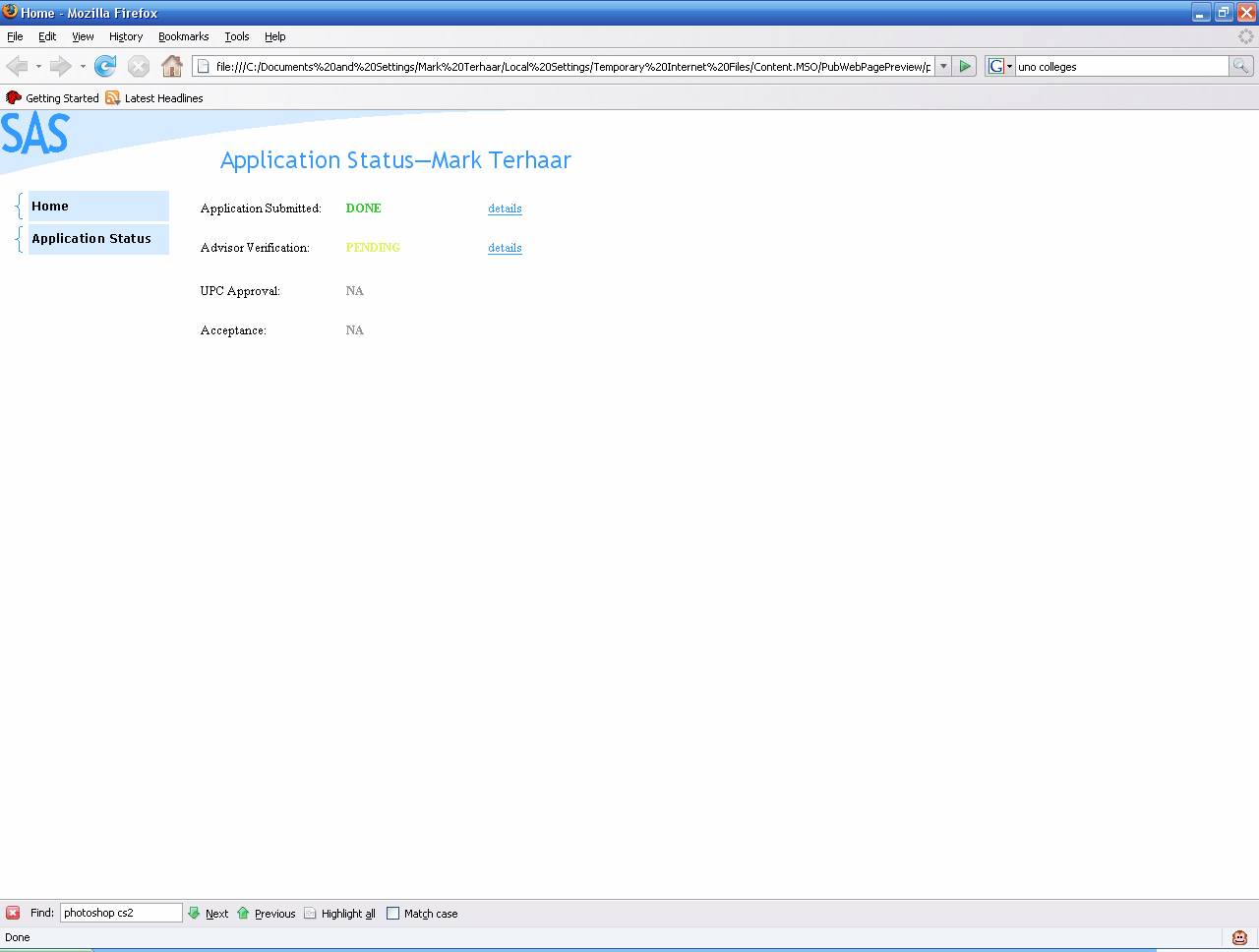


Figure 5 Applicant Status Screen

* + - * 1. **Status Check Controller**

When invoked this controller returns the status of the user’s application.

* 1. ***Data Detailed Design***
     1. **Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| Username | VARCHAR(50) | SAS user’s Username | Primary Key |
| Password | VARCHAR(50) | SAS user’s Password | Not Null |
| First\_Name | VARCHAR(50) | SAS user’s first name |  |
| Last\_Name | VARCHAR(50) | SAS user’s last name |  |
| Address | VARCHAR(50) | SAS user’s address |  |
| City | VARCHAR(50) | SAS user’s city |  |
| State | CHAR(2) | SAS user’s state |  |
| Zipcode | CHAR(10) | SAS user’s zip code |  |
| Phone# | CHAR(13) | SAS user’s phone number |  |
| Email | VARCHAR(50) | SAS user’s e-mail address |  |
| Role | CHAR | SAS user’s Role:  U for Undergrad Applicant  G for Graduate Applicant  R for Registrar  A for Advisor  P for UPC  G for GPC  D for Administrator | Not Null |

* + 1. **Statistics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| #Pending\_Grad\_App | Int | Number of graduate applications that were received but not rejected or accepted yet | Not null |
| #Accepted\_Grad\_App | Int | Number of graduate applications that were accepted. | Not null |
| #Reject\_Grad\_App | Int | Number of graduate applications that were rejected | Not null |
| #Pending\_Undergrad\_App | Int | Number of undergraduate applications that were received but not rejected or accepted yet | Not null |
| #Accepted\_Undergrad\_App | Int | Number of undergraduate applications that were accepted. | Not null |
| #Reject\_Undergrad\_App | Int | Number of undergraduate applications that were rejected | Not null |

* + 1. **Undergraduate Application Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| Undergrad\_Application\_ID | INT | The id for the application | Primary Key |
| Username | VARCHAR(50) | The username of the applicant | Foreign Key |
| First\_Name | VARCHAR(50) | The applicant’s first name |  |
| Last\_Name | VARCHAR(50) | The applicant’s last name |  |
| Address | VARCHAR(50) | The applicant’s Address |  |
| City | VARCHAR(50) | The applicant’s City |  |
| Zip | CHAR(10) | The applicant’s zip code |  |
| SOC# | CHAR(11) | The applicant’s social security number |  |
| Birthday | DATETIME | The birthday of the applicant |  |
| E-mail | VARCHAR(50) | The e-mail address of the applicant |  |
| Phone | CHAR(13) | The applicant’s phone number |  |
| High\_School | VARCHAR(50) | The applicant’s phone number |  |
| High\_School\_GPA | FLOAT | The applicant’s high school GPA |  |
| Is\_Graduated | Int | If the applicant has graduated from high school. 1 = true  0 = false |  |
| Date\_Graduated | DATETIME | The date that the applicant graduated from high school |  |
| High\_School\_Class\_Rank | Int | The applicant’s class rank. |  |
| University\_Program# | Int | A code that represents which program the applicant wants to participate in. |  |
| Planned\_Enrolled\_Date | DATETIME | The date that the applicant wants to be enrolled in the college. |  |
| Criteria1 | Int | If the applicant passed criteria 1.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria2 | Int | If the applicant passed criteria 2.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria3 | Int | If the applicant passed criteria 3.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria4 | Int | If the applicant passed criteria 4.  1 = applicant passed criteria  0=applicant failed |  |
| Criteria5 | Int | If the applicant passed criteria 5.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Advisor\_Notes | VARCHAR(250) | This is notes that the advisor wrote about the application |  |
| Application\_Accepted | Int | This is where the UPC approves or rejected the applicanton  1= approved  2=rejected |  |
| Application\_Finished | Int | Is used by the system when the SAS is finished processing the application. (Letter sent out, data sent to the admission and student information system)  1 = finished  0 = unfinished |  |

* + 1. **Graduate Application Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| Grad\_Application\_ID | INT | The id for the application | Primary Key |
| Username | VARCHAR(50) | The username of the graduate applicant | Foreign Key |
| First\_Name | VARCHAR(50) | The applicant’s first name |  |
| Last\_Name | VARCHAR(50) | The applicant’s last name |  |
| Address | VARCHAR(50) | The applicant’s Address |  |
| City | VARCHAR(50) | The applicant’s City |  |
| Zip | CHAR(10) | The applicant’s zip code |  |
| SOC# | CHAR(11) | The applicant’s social security number |  |
| Birthday | DATETIME | The birthday of the applicant |  |
| E-mail | VARCHAR(50) | The e-mail address of the applicant |  |
| Phone | CHAR(13) | The applicant’s phone number |  |
| High\_School | VARCHAR(50) | The applicant’s phone number |  |
| High\_School\_GPA | FLOAT | The applicant’s high school GPA |  |
| Is\_Graduated | Int | If the applicant has graduated from high school. 1 = true  0 = false |  |
| Date\_Graduated | DATETIME | The date that the applicant graduated from high school |  |
| High\_School\_Class\_Rank | Int | The applicant’s class rank. |  |
| Undergrad\_College | VARCHAR(50) | The name of the college that applicant graduated from |  |
| Grad\_GPA | FLOAT | The GPA from the college applicant graduated from |  |
| Grad\_Degree | VARCHAR(50) | The name of the degree that the applicant received after graduating from college |  |
| Grad\_Graduation\_Date | DATETIME | The date that the applicant graduated from high school |  |
| University\_Program# | Int | A code that represents which program the applicant wants to participate in. |  |
| Planned\_Enrolled\_Date | DATETIME | The date that the applicant wants to be enrolled in the college. |  |
| Criteria1 | Int | If the applicant passed criteria 1.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria2 | Int | If the applicant passed criteria 2.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria3 | Int | If the applicant passed criteria 3.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria4 | Int | If the applicant passed criteria 4.  1 = applicant passed criteria  0=applicant failed |  |
| Criteria5 | Int | If the applicant passed criteria 5.  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria6 | Int | If the applicant passed criteria 6  1 = applicant passed criteria  0=applicant failed criteria |  |
| Criteria7 | Int | If the applicant passed criteria 7.  1 = applicant passed criteria  0=applicant failed criteria |  |
| GPC\_Notes | VARCHAR(250) | This is notes that the GPC wrote about the application |  |
| Application\_Accepted | Int | This is where the GPC Chair approves or rejected the application  1= approved  2=rejected |  |
| Application\_Finished | Int | Is used by the system when the SAS is finished processing the application. (Letter sent out, data sent to the admission and student information system)  1 = finished  0 = unfinished |  |

* + 1. **Graduate Relation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| Relation\_ID | INT | An unique ID for the relation | Primary Key |
| Username | VARCHAR(50) | The username for the Person evaluating the graduate applicant | Not null |
| Grad\_Applicant\_ID | INT | The application that the person with the username is working on | Not null |

* + 1. **Undergraduate Relation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Description** | **Constraints** |
| Relation\_ID | INT | An unique ID for the relation | Primary Key |
| Username | VARCHAR(50) | The username for the Person evaluating the undergraduate applicant | Not null |
| Grad\_Applicant\_ID | INT | The application that the person with the username is working on | Not null |