**Student Design Network:**

**Object Design Document**

**1. Introduction              3**

**1.1 Object design trade-offs              3**

**1.2 Interface documentation guidelines              3**

**1.3 Definitions, acronyms, and abbreviations              3**

**1.4 References              3**

**2. Packages               3-4**

**3. Class Interface              4-6**

**1. Introduction**

**1.1 Object design trade-offs**

The trade off of our student social networking site is between modularity/reusability and efficiency/cost. We wanted to design classes as modular as possible. However, this would result in inefficiencies due to function call overload and cost more because of the time spent. Therefore, we needed balance these two groups of properties. We applied modularity and reusability principles to wherever suitable but we also tuned critical functions for efficiency.

**1.2 Interface documentation guidelines**

There is one big package defined in the object design model of the student portal named STUDENT DESIGN NETWORK Classes. Since all the user interfaces will be web pages, there are not any classes created for the user interface

In the names of classes in the STUDENT DESIGN NETWORK Classes package, attributes and methods of these classes, clear naming conventions are used.  The names of each object, attribute or method will be understandable by looking at the name.

**1.3 Definitions, acronyms, and abbreviations**

**RAD   :** Requirements Analysis Document

**SDD   :** Software Design Document

**ODD  :** Object Design Document

**UML   :** Unified Modeling Language

**1.4 References**

Object-Oriented Software Engineering Using UML, Patterns, and Java Second edition, Bernd Bruegge – Allen H Dutoid, Prentice Hall International, 2004

**2. Packages**

STUDENT DESIGN NETWORK Classes Package consists of the Database Subsystem and the StudentSocialSoftware Management Tool. These are the subsystems, which are implemented by AII Space.  The StudentSocialSoftware Management Tool interacts with the database.

The StudentSocialSoftware Management Tool manages the functions of AII Space.

StudentSocialSoftware Management Tool is broken down into 3 packages.

∙                Interface

∙                Application Logic

∙                Storage

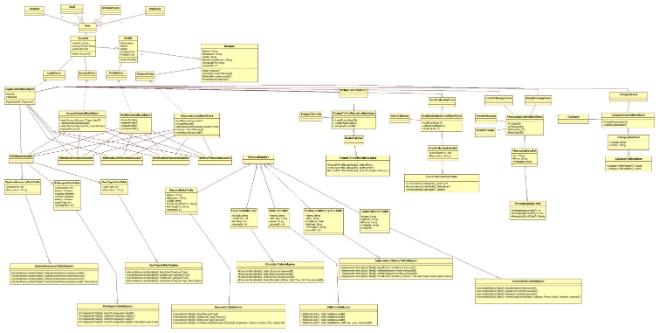
The Interface package includes all of the GUI information, which is what the users see and how they interact with STUDENT DESIGN NETWORK.

The Application Logic Package holds all of the classes that STUDENT DESIGN NETWORK uses to function. It contains all of the resources to transmit data to other classes within the package. This is where all of the logic is to run all of the functions. (Ex, Update Resume, Print Resume…)

The Storage Packing simply contains all of the classes used to store information on the database.

The Database holds all of the data for all the users and talks directly to the StudentSocialSoftware Management Tool.

**3. Class Interfaces**

****

**Account**  -  The Account class sends form information to the Account Control Bus Object .

**Profile** - The Profile class sends form information to the Profile Control Bus Object.

**Resume** - The Resume class sends form information to the Resume Control Bus Object.

**AccountControlBusObject** - The Account Control Bus Object class inserts the user profile information into the database.

**Profile** **ControlBusObject** - The Profile Control Bus Object class inserts the user profile information into the database.

**Resume** **ControlBusObject** - The Resume class inserts the user resume information into the database.

**LoginControlBusObject** – Uses the information from the data table to confirm a user so they can access their account.

**SystemRecourceDataTable** -   Container for holding System resource data.

**PriviligesDataTable** – Container for holding Privileges Data

**UserDataTable** – Container for holding User Data.

**ResumeDataTable** – Holds all of the resume information.

**EducationDataTable** – Container for holds all of the Education data.

**SkillsDataTable** – Container for holding skill data.

**ContactInfoDataTable** – Container for holding Contact data.

**EmploymentHistoryDataTable** –

**PictureToProfileDataTable -** Container for holding Picture Profile.

**EventCalendarDataTable -** Container for holding Event Calendar.

**MessageControlDataTable -** Container for holding Message Control.

**CategoryDataTable -** Container for holding Master Category.

**SystemRecourceTableAdaptor** - Holds the system resource information from the database.

**UserTypeTableAdaptor** - Holds the user type information data of the User types from the database.

**EducationTableAdaptor** - Holds the education information data filtered from the resume data table.

**EmploymentHistoryTableAdaptor** - Holds the Employment History information data filtered from the resume data table.

**PriviligesTableAdaptor**- Contains all of the privilege information.

**ResumeTableAdaptor** - Holds all of the resume data and insets it into the database.

**SkillsTableAdaptor**- Holds the skills data filtered from the resume data table.

**ContactInfoTableAdaptor** - Holds the contact info data filtered from the resume data table.

**PictureToProfileTableAdaptor -** Holds the picture profile number info data filtered from the PictureToProfile data table.

**EventCalendarTableAdaptor -** Holds the event calendar info data filtered from the event calendar data table.

**MessageControlTableAdaptor -** Holds the message info data filtered from the message data table.

**CategoryTableAdaptor -** Holds the master category info data filtered from the master category data table.

**ProfileForm –** Profile Form sends form information to the Profile Bus Object .

**ResumeForm-** sends form information to ResumeControlBusinessObject.

**ResumeControlBusinessObject** **–** sends information to the Privilidges DataSet.

**EventCalendar** – EventCalendar class sends form information to the

**EventCalendarForm** – Holds all the calendar

**EventCalendarControlBusinessObject-** sends information to the Privilidges DataSet.

**PictureToProfile** – PictureToProfile sends information to the PictureToProfileForm into the database

**PictureToProfileForm –** PictureToProfileForm holds the PictureToProfile information data

**PictureToProfileControl-** PictureToProfileForm object will send the information to

**CreateMessage** – sends information to the Messa

**CreateMessageForm -** sends form information to MessagingControlBusinessObject

**ReplyMessage** **–** Reply Message sends form information to the Messaging Control Business Object.

**ReplyMessageForm -** sends form information to Messaging ControlBusinessObject

**MessagingControlBusinessObject -** sends information to the Privilidges DataSet.

**Category** **–** Create Master Category sends the information to the Create Master Category Control Business Object

**CreateMasterCategoryForm -** sends form information to CreateMasterCategoryControlBusinessObject

**CategoryControlBusinessObject** - sends information to the Privilidges DataSet.