

UTB Accreditation Software

Software Design Document

Version 1.0

**Prepared for
Software Engineering
Instructor: MK Quweider, Ph.D.**

**Igor Gnatchenko
Andrey Gnatchenko
November 10 , 2014**

Fall 2014

Revision History

Revision 1

11/10/14

Table of contents

1 Introduction	3
1.1 Purpose	3
1.2 Scope.....	3
1.3 Overview.....	3
1.4 Definitions and Acronyms	3
1.4 Customer	4
2 System Overview	4
3 System Architecture	5
3.1 Subsystem decomposition	5
3.2 Class diagram.....	7
3.3 Sequence diagram.....	10
3.4 State diagram.....	17
3.5 Activity diagram.....	19
3.6 Entity Relationship diagram	21
3.7 GUI design.....	23
4 Work schedule	28

1 Introduction

1.1 Purpose

This document contains detailed explanation of the design of the UTBAS application. It can be used by the developer team in order to code this software, by the test teams that will test the application. It describes the architecture and the main structure of the UTBAS application.

1.2 Scope

The UTBAS application allows users to do annual accreditation work within the university, create reports based on specific data and provide useful mechanism to store all the related information. It also provides the means by which a user is able to manipulate data stored in the application database and be notified about the changes in the system.

The application has useful graphical user interface which helps users to work with data easily. This data can be represented on the screen in form of buttons, labels, icons, tables and other graphical components. Also, one of the benefits that this application has is that this application manages all the data related to accreditation process and allows users to control this process from one central place.

1.3 Overview

The main idea of this document is to give detailed explanation on how UTBAS application will work. The document is organized into separated sections and each section describes a part of the application. Also, this document is enriched with graphical material that will help a reader to better understand structure of the application.

1.4 Definitions and Acronyms

Term	Definition
GUI	Graphical User Interface
UTB	University of Texas at Brownsville
SRS	Software Requirements Specification
IDE	Integrated Development Environment
Database	A special structure to store information
CVS	Concurrent Version System
UML	Unified Modeling Language
Email	Electronic Mail
Java	Object-oriented programming language
PDF	Portable Document Format
POS	Program of study
CV	Curriculum vitae

JVM	Java Virtual Machine
RAM	Random Access Memory
CPU	Central Processing Unit
OS	Operating System
UTBAS	University of Texas at Brownsville Accreditation System

1.4 Customer

The potential customers of the application are teachers that can use this application to do accreditation work or university authorities that can utilize this accreditation system in order to create reports related to the university.

2 System Overview

The application can provide a user with a wide range of functions. These functions let the user to do accreditation work related to universities, colleges and departments with ease. These functions can be written as follows:

Provide a user with the installation wizard

Show a splash screen when a user starts the application

Give access in two modes - the admin mode and the user mode

Input university information

- The name and the logo of the university
- The mission and the vision of the university
- The number of colleges belonging to the university

Input college information

- The name and the logo of the college
- The mission and the vision of the college
- The number of departments belonging to the college

Input department information

- The name and the logo of the department
- The mission and the vision of the department
- The number of learning programs belonging to the department

Input student information

- The first and the last name of the student
- The score of the student
- The gender of the student
- The age of the student

Input course information

- The name of the course
- The description of the course
- The number of practical lessons
- The number of lectures

Output selected element information

- Create a new element
- Edit an existing element
- Delete a selected element
- Print information about selected elements
- Notify users about changes via email
- Change display settings
 - Font settings
 - Color settings
 - Theme settings
- Show hits if needed
 - Popup hits
 - Information labels
 - Embedded help

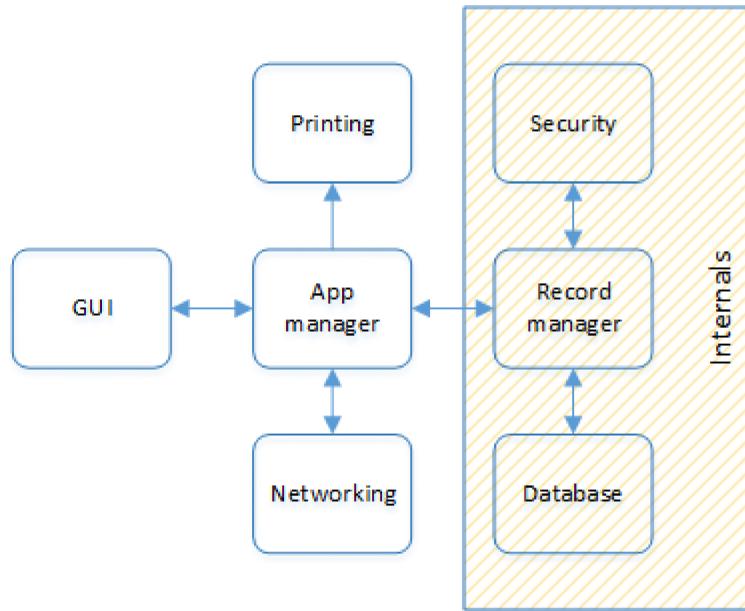
3 System Architecture

3.1 Subsystem decomposition

The application consists of many modules and, thus, a bit complex. To make it easier to understand the relations between subsystems and modules, it is advisable to separate the whole application into meaningful pieces or parts, each of which can bear a specific function. The application is divided into the following modules:

- Network module
- Validation module
- GUI module
- Database module
- Security module
- Printing module

Modules are represented as subsystems. Module decomposition and main relations between subsystems are illustrated on the next figure.



Subsystem decomposition

Networking module

The networking module is responsible for sending Internet based notifications to a user in order to notify him about the changes that took place. This module will utilize functionality of Java libraries. Interface between the networking module and other modules is maintained in such a way that it allows developers to switch between different versions of given Java libraries so as to facilitate robustness of the application.

Validation module

The security module provides authorization and validation functions. It is possible to implement this module because there is a variety of external libraries that provide necessary functional basis. The security module itself will play a role of an interface to the given functionality wrapping the library's methods in a class. It will facilitate project decoupling and help to keep code structures in a tidy manner.

GUI module

The GUI interface is the best approach when it comes to deal with user interaction and user input. For the most applications it is critical to implement convenient interface when a user feels ease working with an application. The most disappointing thing that can happen is that the application suits all functional demands, but does not meet requirements in terms of convenience. The implemented interface will be controlled by events that are generated every time when the user presses the button or clicks a menu option.

Database module

The database module provides functional methods to store, retrieve and maintain application data in an internal relational database. It is meaningful to utilize the database in the current applications, because it allows the system to manipulate data with ease provided by SQL-based commands, which are fast, reliable and secure.

Security module

Implementation of the security module enables the system to work into different modes: common and augmented. The common mode (user mode) provides only typical operations to the user. In contrast, the augmented mode (administrator mode) gives the user powerful features to work with the application. It is important to notice that the user will be prompted to enter a password when he is trying to get access to the augmented mode. It is done for security reasons.

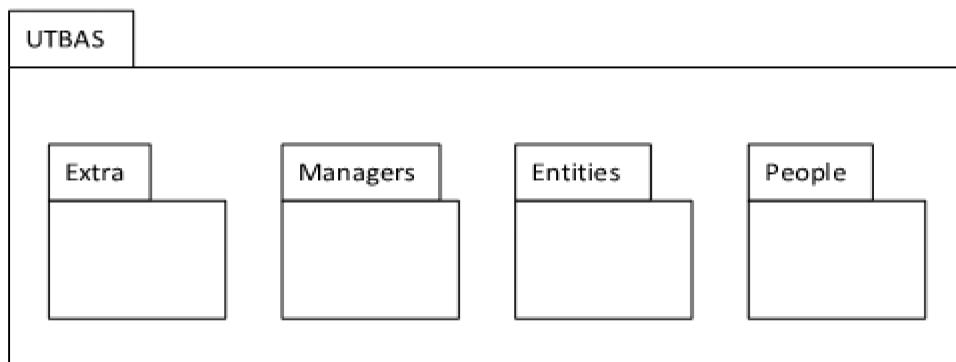
Printing module

The printing module comes in when it is necessary to create a good-looking report which contains all the information that was selected by a user. This module should have functionality that will allow the application to retrieve object data, create an appropriate and valid XML document, convert this document into a HTML format and, finally, create a PDF document based on this information.

3.2 Class diagram

Class diagram shows static relations between objects in the application. It can be very useful when the development team implements main classes of the application. Each class belongs to a certain package. It makes the structure of classes clear and understandable.

The main package structure is shown on the next illustration. The main package, which is UTBAS, consists of four sub packages: Extra, Managers, Entities and People. Each of them will be described in more details later as well as the relations between them.

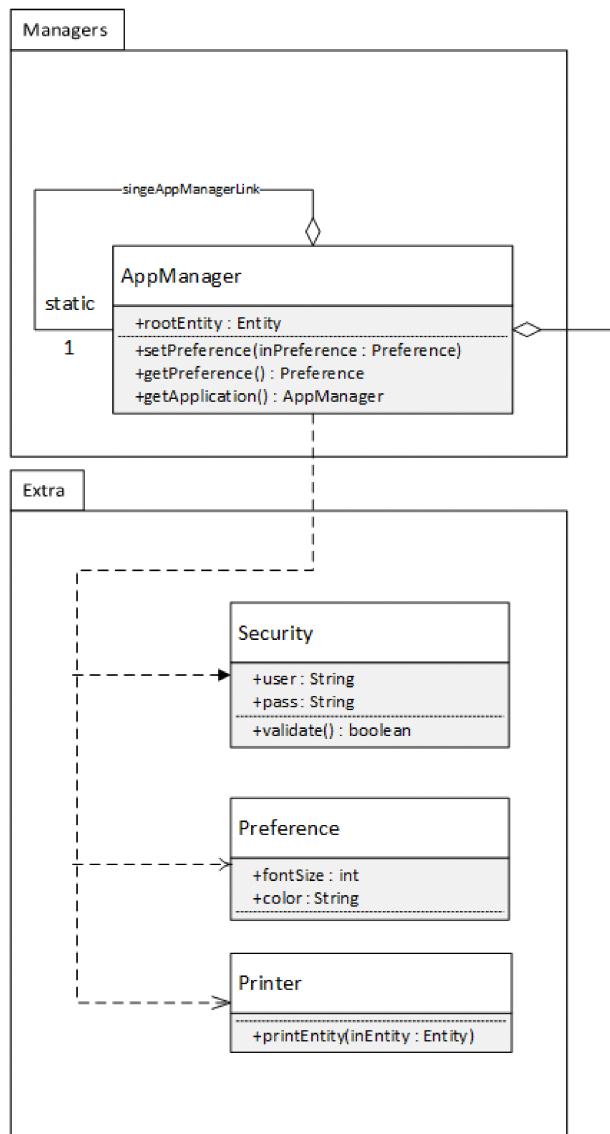


Package structure of the application

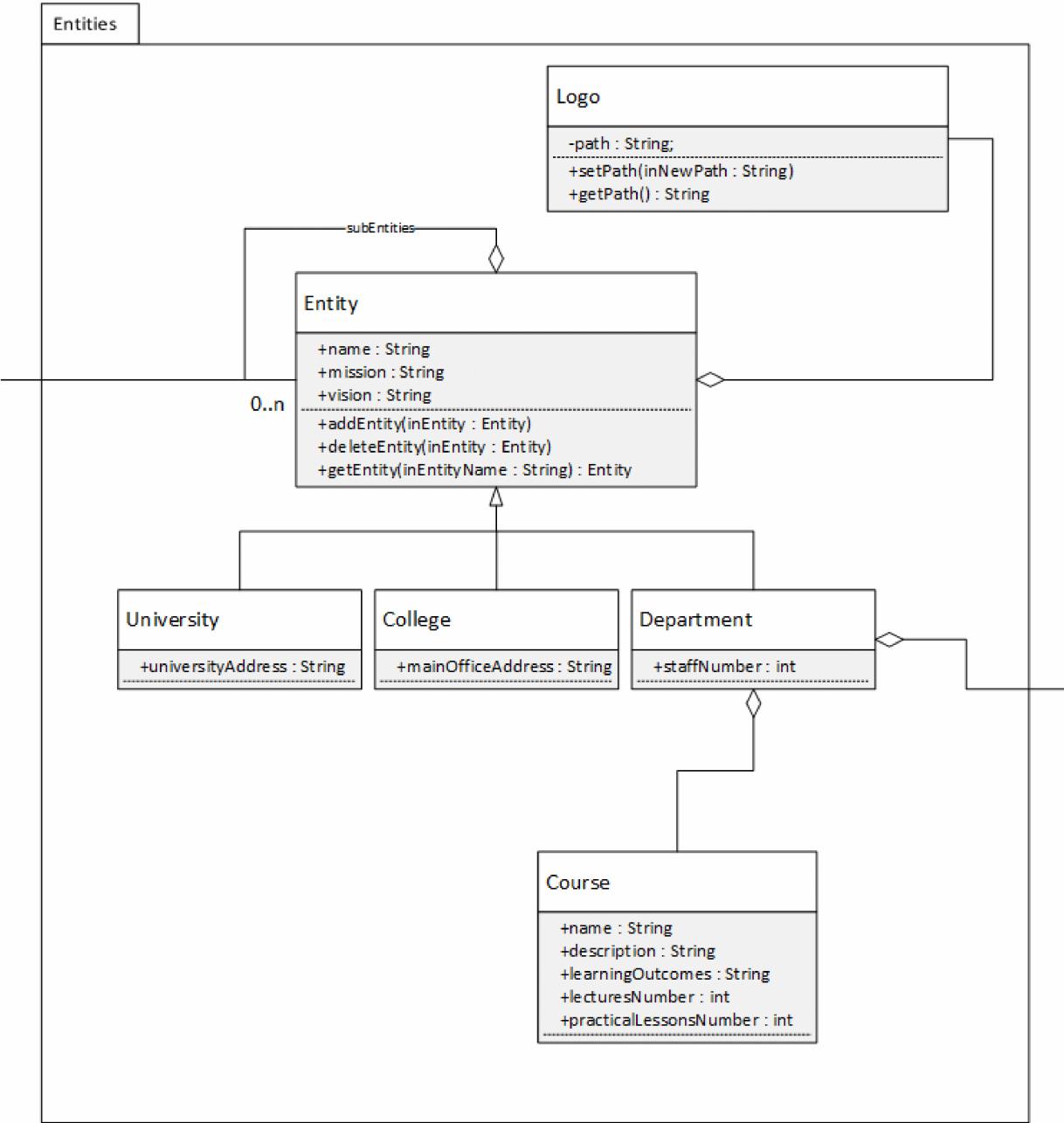
The Managers package contains general classes that control the entire application. Application preferences and the root element, which represents database records, will be stored in the AppManager class. Also, the Extra package focuses on extra functionality that the application will provide to its users. Hence, Security, Preference and Printer classes are stored here.

The Security class encapsulates validation procedures. Whenever a user wants to get administrative privileges he needs to go through the validation process. It means that he needs to know the administrator's password and login. The Preference class helps to adjust colors and fonts used the application.

The application is designed in such a way that it is customizable, a user can set the color with which elements will be drawn on the screen. Also, he can specify the size of the font. The Printer class is in charge of creating PDF reports. A user can select any element, press a button and the application will form a report containing information related to this element.



The Managers and The Extra packages

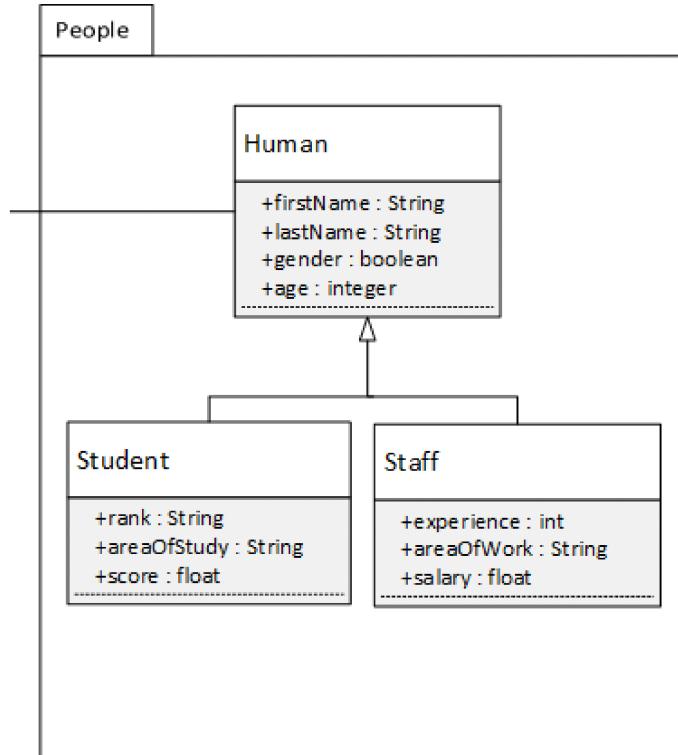


The Entities package

The Entities package consists of classes that mainly deal with structural elements of the application. It means that each class that is located in the Entities package has to go about certain aspect of the hierarchy of data stored in the database. The Entity class deserves thorough description. It serves as a main entry point in the hierarchy.

The internals of the class are organized in such a way that allows the application to treat all entities of different type in a similar way. Each of them has a link to the next element in the hierarchy and is bound to the element of a higher level. It is very useful to keep links to the similar elements when it comes to working with the whole hierarchy of elements. For instance, when a

user wants to print or display certain part of hierarchy it can be easily done by means of recursion.

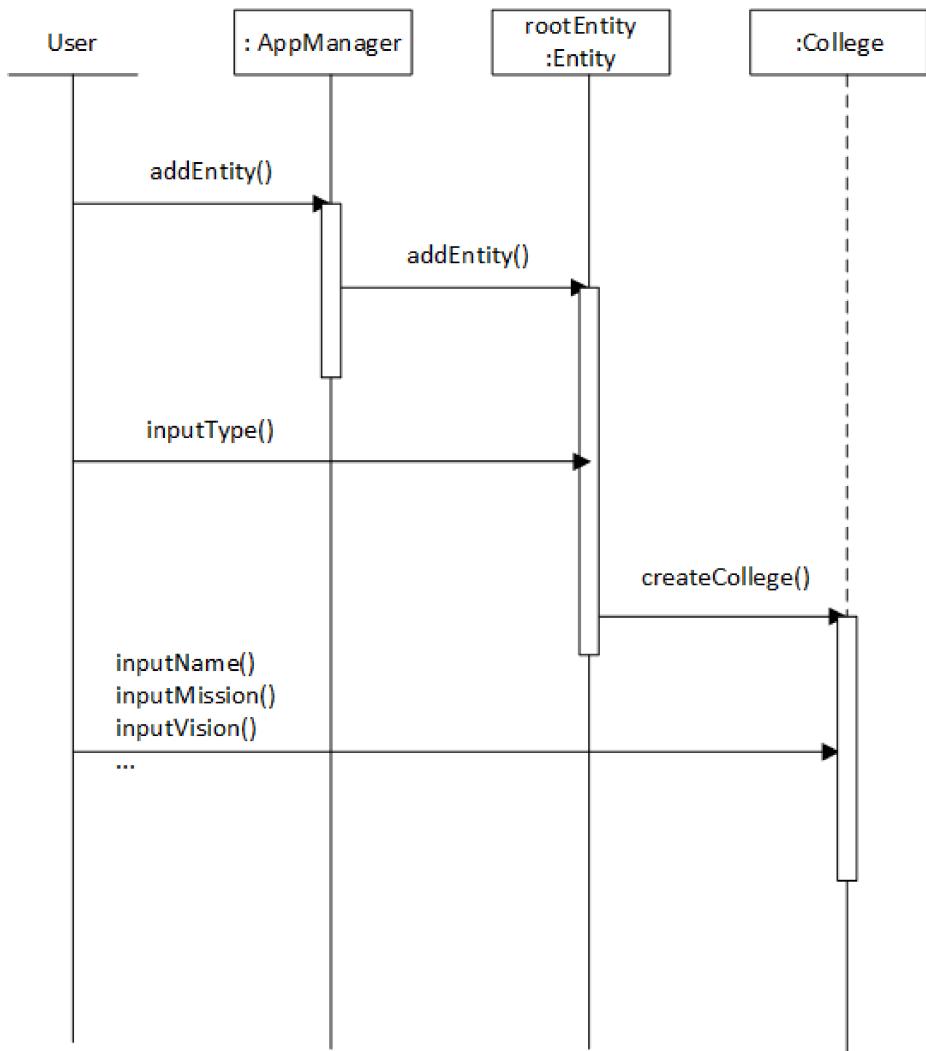


The People package

The last package contains human related classes. There is a class which aggregates the common characteristics of the remaining . It is important to keep a mutual part of classes in a separate parent class, because it facilitates flexibility and maintainability of the derived classes.

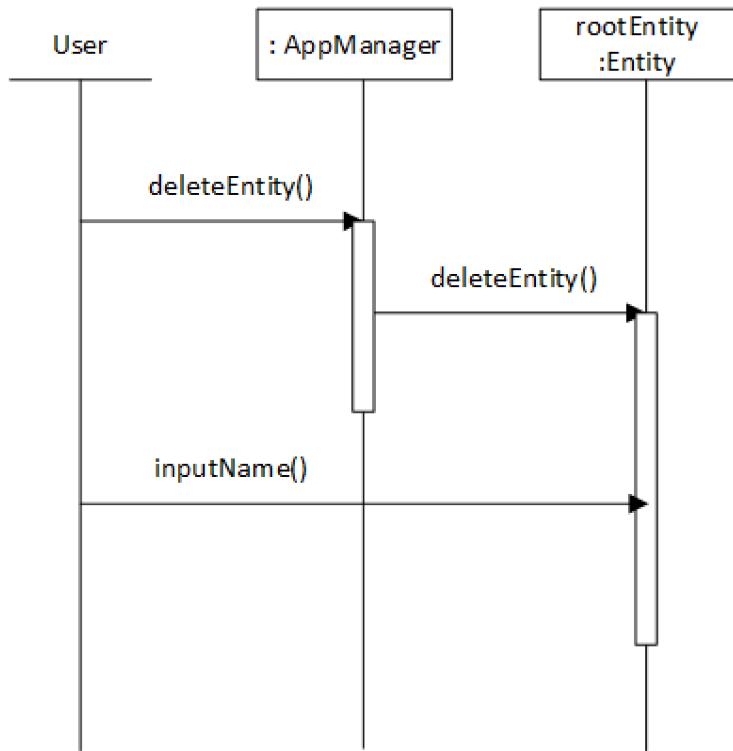
3.3 Sequence diagram

There are many different sequence diagrams presented in this document. Each of these diagram shows the sequence of actions and can be used when analyzing the behavior of classes. Sequence diagrams show dynamic relations between classes, meanwhile class diagrams represent a static picture.



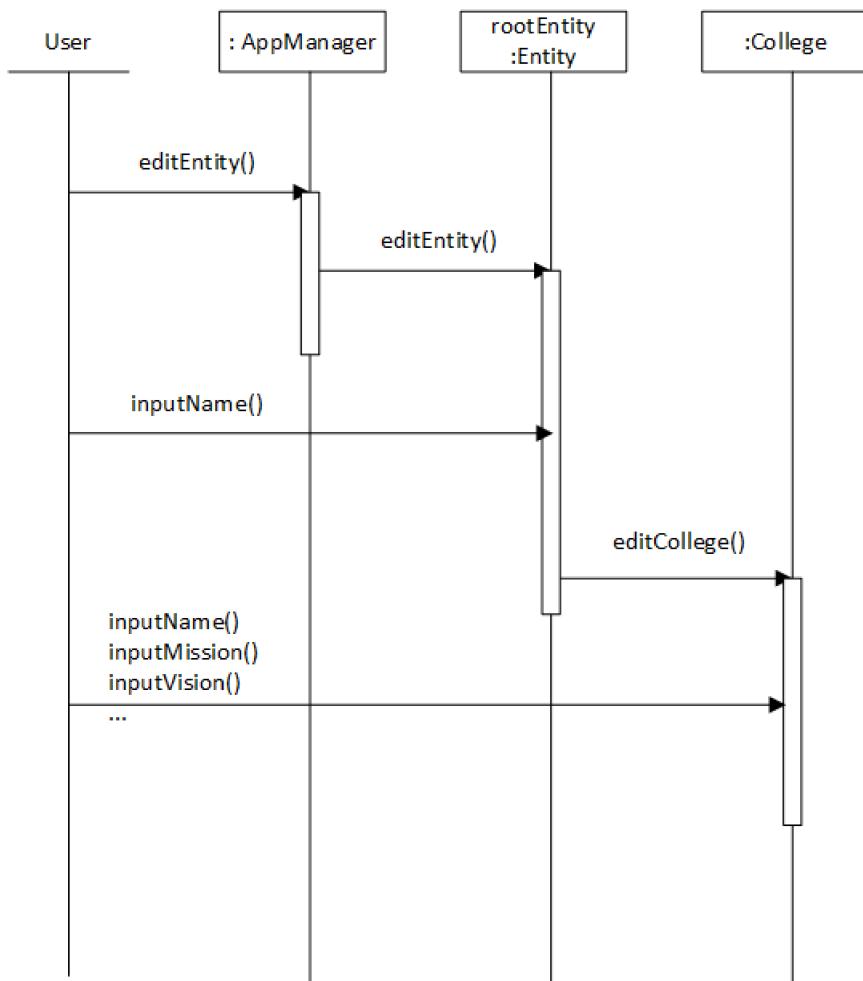
Add Entity sequence diagram

Add Entity sequence diagram depicts the process of adding one of the entities to the database with which the application can work later. The important thing to notice is that diagram remains very similar in different situations. An entity the application operates with is a general class that collects mutual characteristics of its subclasses, thus, increasing the reusability and reducing the size of program code.



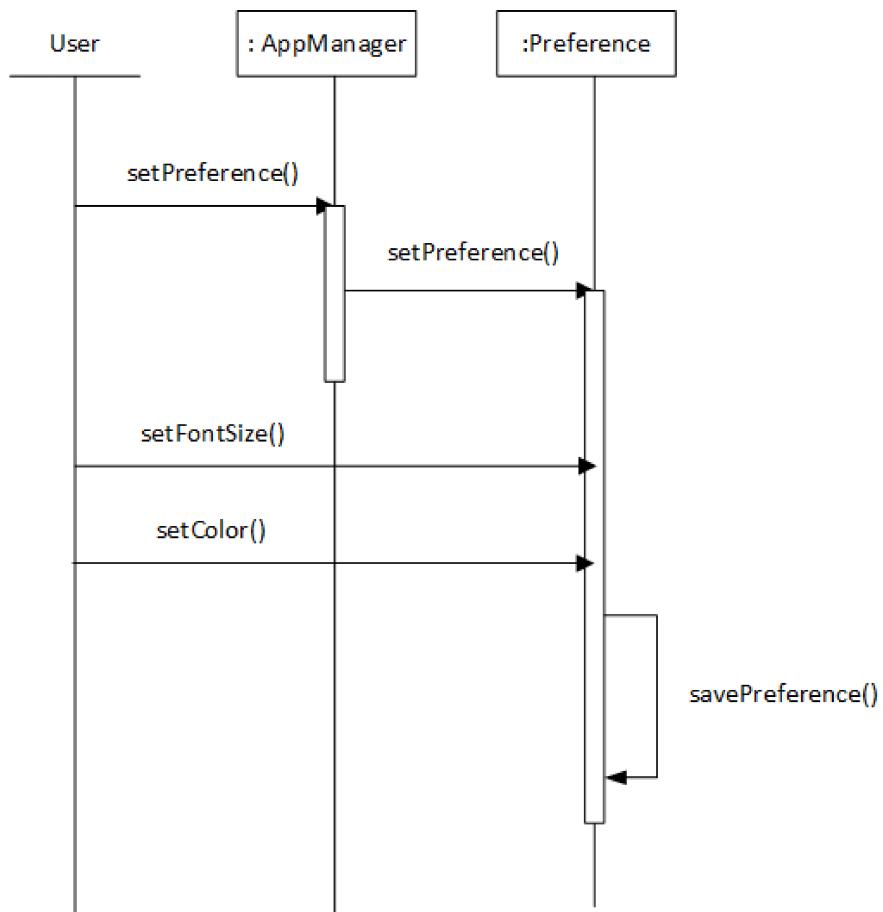
Delete Entity sequence diagram

The main idea of Delete Entity diagram is to show the process of deleting an entity from the application. The entity that is going to be deleted can be specified by passing the name of this entity to the `deleteEntity` function. Here again, there is no need to show the same diagram many times as this one is applicable for all three cases - University class, College class and Department class.



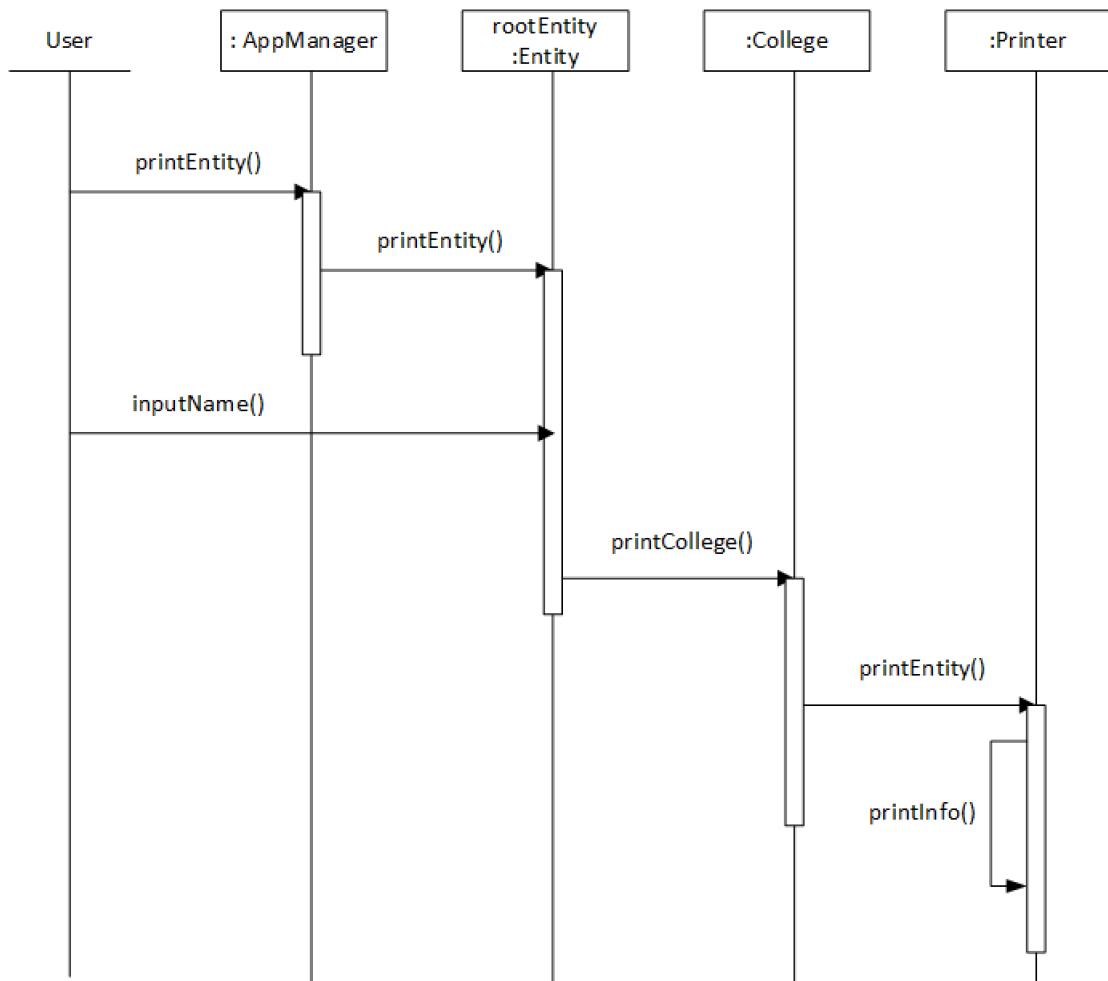
Edit Entity sequence diagram

When a user wants to edit the entity he has chosen, he can do so by clicking special button, which says "Edit". This sequence diagram shows what happens in this case.



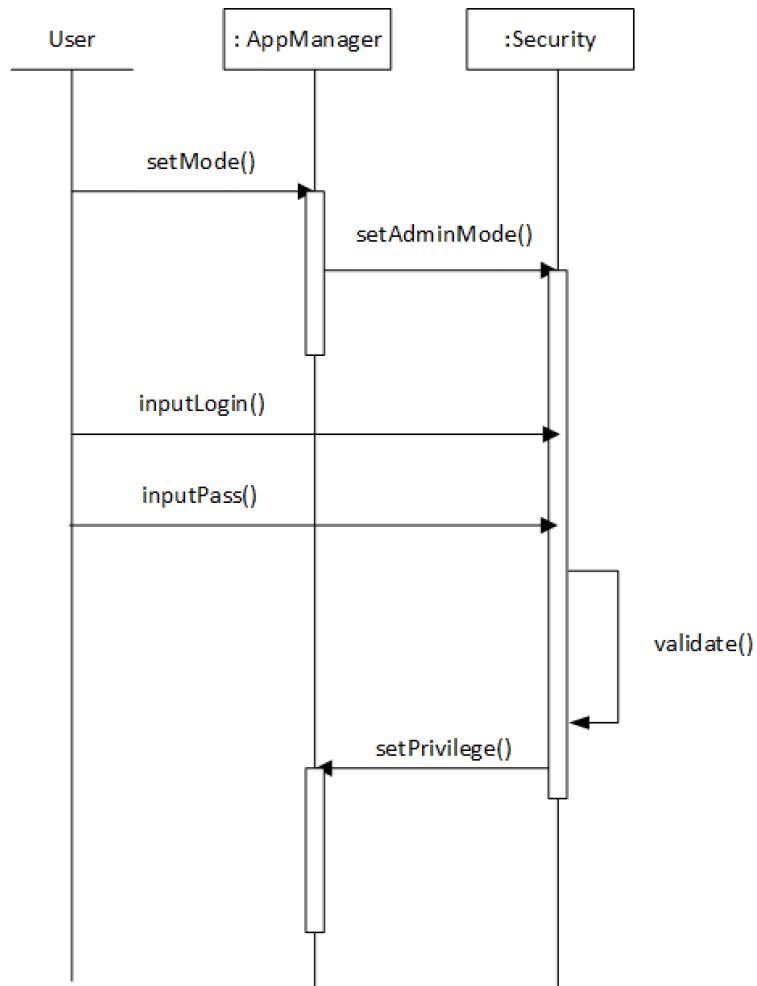
Preference sequence diagram

Preferences are extremely useful technique that helps a user to save some graphic options, the look and feel of the application etc. In case of this project preferences are stored in the Preference class. As the diagram illustrates, the user can set the color of GUI as well as the size of the text.



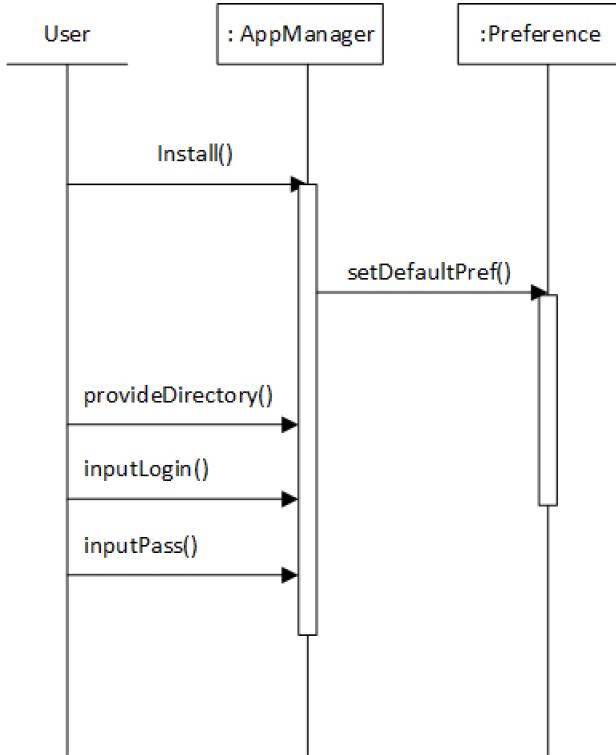
Printing Entity sequence diagram

When it comes to printing, much work is done without user's notion. A user can just click a corresponding button and get a newly created PDF document. By looking on Printing Entity diagram it becomes clear that before the user can get his document, the data should be retrieved and the printing algorithm should continue its flow. A more detailed explanation on how a PDF document can be formed is shown in Activity diagrams.



Set Mode sequence diagram

A user wants to be able to add, modify and delete certain records from the application. At the same time, there certainly should be some protection mechanism implemented. This sequence diagram describes how the validation of users can be done.

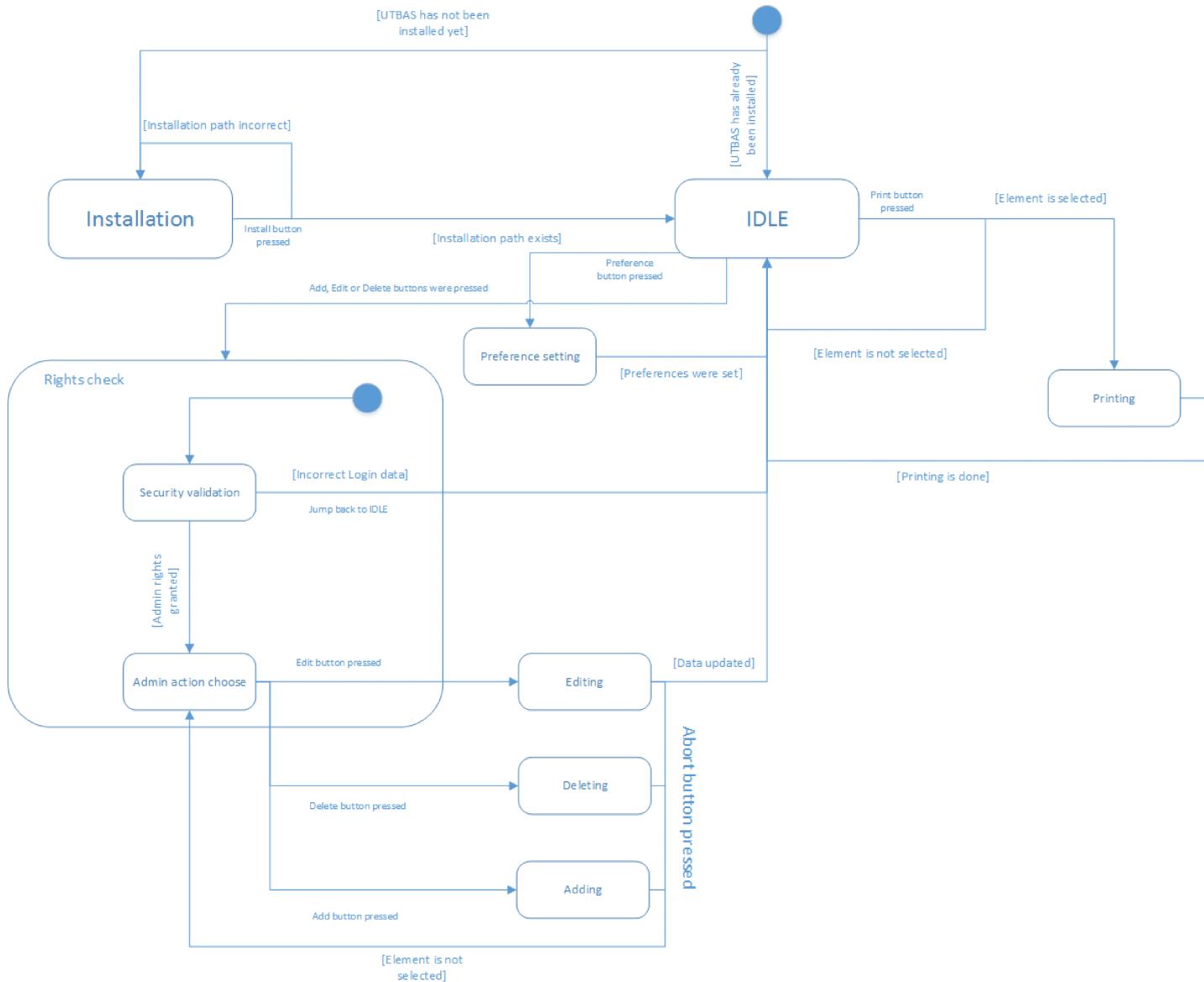


Wizard Sequence diagram

Wizard diagram illustrates the usual sequence of steps that a user should go through in order to install the application. It should be noticed that the user must specify the appropriate system path in which the application should be unpacked. Also he is supposed to input initial the administrative login and password.

3.4 State diagram

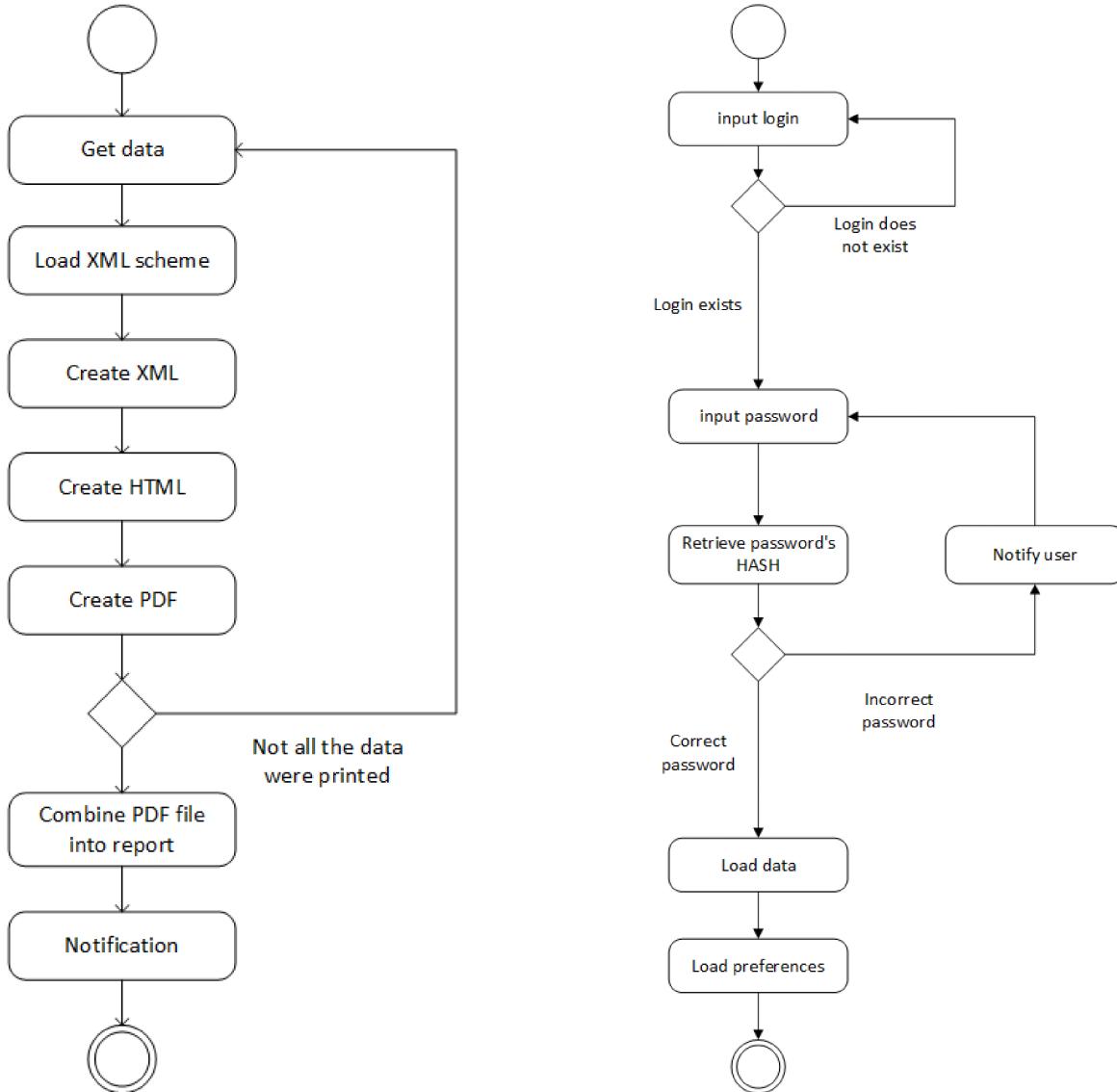
State diagrams are widely used when a developer wants to describe the GUI interface of the application. Following this idea, he represents an entire application by many states that are connected with each other. The following diagram shows states and transitions between them in UTBAS application.



State diagram

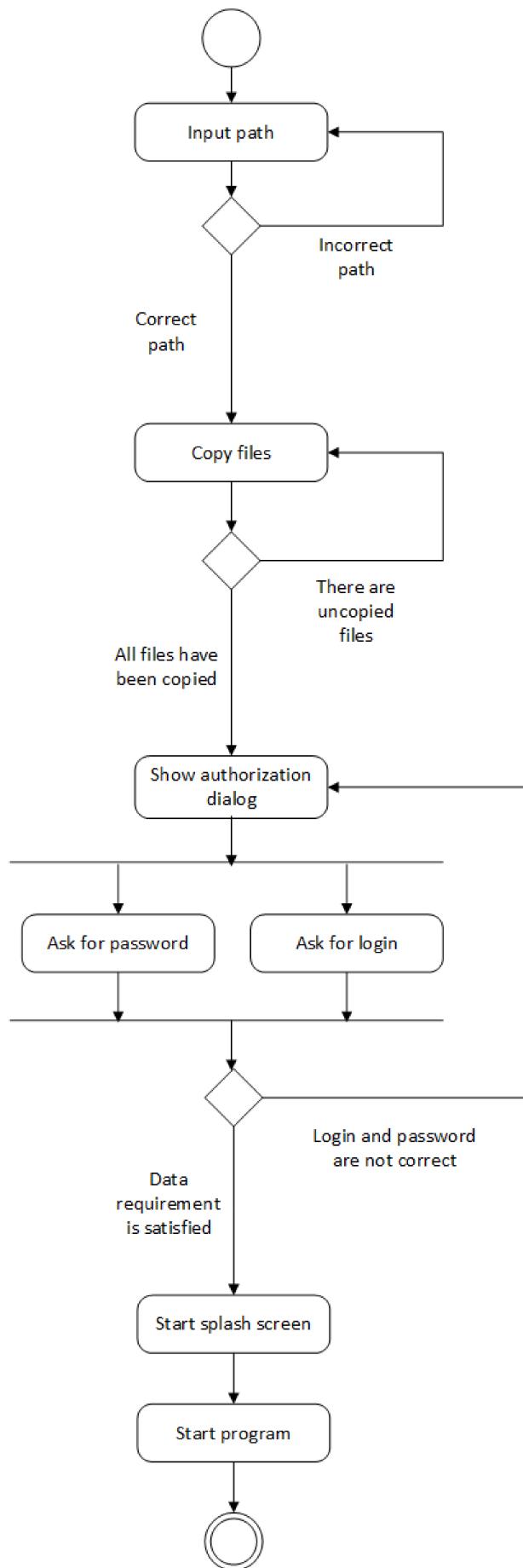
3.5 Activity diagram

Activity diagrams are used to show the exact order of operations according to which some work should be done. The following activity diagrams show printing, verification, installation and XML document creation procedures in details.

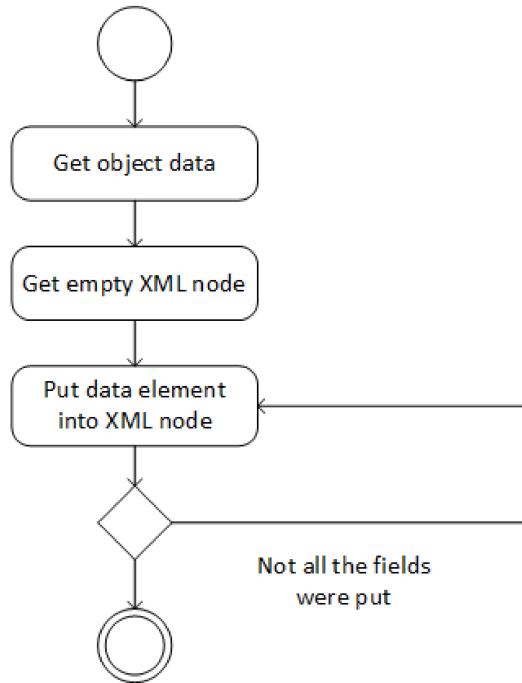


Printing Activity and Verification Activity diagrams

A few words should be said about Verification diagram. It uses hashes of passwords as this approach is better in terms of security. Passwords are not stored as plaintext. Only hashed are available.



Wizard Activity diagram



XML Creation Activity diagram

3.6 Entity Relationship diagram

The application provides functionality to work with the internal database. The entire structure of the database and links between tables are illustrated with help of Entity Relationship diagram. The root element (the main table) of the database is the university table. This table encapsulates the most common information about the university and its parts.

It is important to notice that the university table maintains external links by means of a primary key. At the same time, the same field is ingrained in the structure of the college table. It means that each time the application is trying to retrieve information about a certain college that belongs to the university, it should retrieve only those colleges whose ID_UNIVERSITY is the same as ID of the main university. The same approach is utilized when a user wants to get information about departments belonging to a certain college.



Entity Relationship diagram

Each department has links to the students who is currently studying, staff that are working and the courses that the department can offer. What is following is the detailed description of all the tables.

University		
Field name	Type	Description
ID	int	Primary key of the table
name	varchar (256)	Name of the university
mission	varchar (2048)	Mission of the university
vision	varchar (2048)	Vision of the university
address	varchar (512)	Address of the university
logo	varchar (128)	Path to the logo icon

College		
Field name	Type	Description
ID	int	Primary key of the table
ID_UNIVERSITY	int	Foreign key for the University table
name	varchar (256)	Name of the college
mission	varchar (2048)	Mission of the college
vision	varchar (2048)	Vision of the college
mainOfficeAddress	varchar (512)	Address of the main office of the college
logo	varchar (128)	Path to the logo icon

Department		
Field name	Type	Description
ID	int	Primary key of the table
ID_COLLEGE	int	Foreign key for the College table
name	varchar (256)	Name of the department
mission	varchar (2048)	Mission of the department
vision	varchar (2048)	Vision of the department
staffNumber	int	Number of staff
logo	varchar (128)	Path to the logo icon

Course		
Field name	Type	Description
ID	int	Primary key of the table
ID_DEPARTMENT	int	Foreign key for the Department table
name	varchar (256)	Name of the course
description	varchar (2048)	Description of the course
learnOuts	varchar (2048)	Learning outcomes of the course
lectures	int	Number of lectures
practices	int	Number of practical lessons

Staff		
Field name	Type	Description
ID	int	Primary key of the table
ID_DEPARTMENT	int	Foreign key for the Department table
firstName	varchar (64)	First name
lastName	varchar (128)	Last name
gender	boolean	Male is 1, female is 0
age	int	Age
experience	int	Number of years working
areaOfWork	varchar(512)	Specialty
salary	float	Salary (in USD)

Student		
Field name	Type	Description
ID	int	Primary key of the table
ID_DEPARTMENT	int	Foreign key for the Department table
firstName	varchar (64)	First name
lastName	varchar (128)	Last name
gender	boolean	Male is 1, female is 0
age	int	Age
rank	varchar (64)	Is he a graduate, undergraduate or postgraduate student?
areaOfStudy	varchar(512)	Major subject
score	float	GPA score

3.7 GUI design

In this project, the development team decided to implement GUI interface for a verity of reasons. GUI interface is intuitive, easy to understand and there are many libraries that can help to build resizable and robust GUI interface. The GUI interface of UTBAS application consists of many panels. Each panel has its own purpose. Also, the links between different GUI component are reduces to a minimum, because it would be easy to upgrade a certain part of GUI interface in case of necessity. The following diagrams show how the GUI for UTBAS application will be organized.

New Entity

Name	Provided name
Mission	Provided name
...	
Group	Group name
Param_n	Provided value

Create Entity dialog

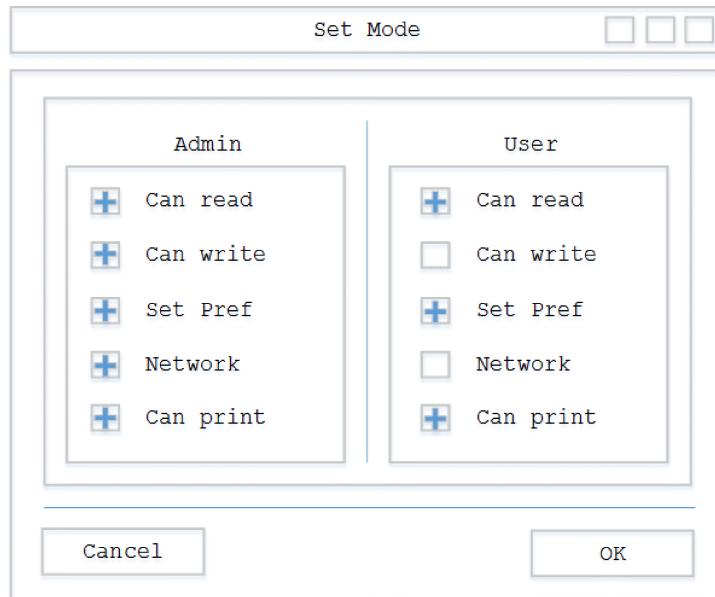
When creating a new entity, a user is prompted to enter specific information related to this entity, such as the "Name" and the "Mission". The Dialog can be closed by clicking the "Cancel" button.

Entity Name

Name	Provided name
Mission	Provided name
Vision	Provided name
...	
Param_n	Provided value

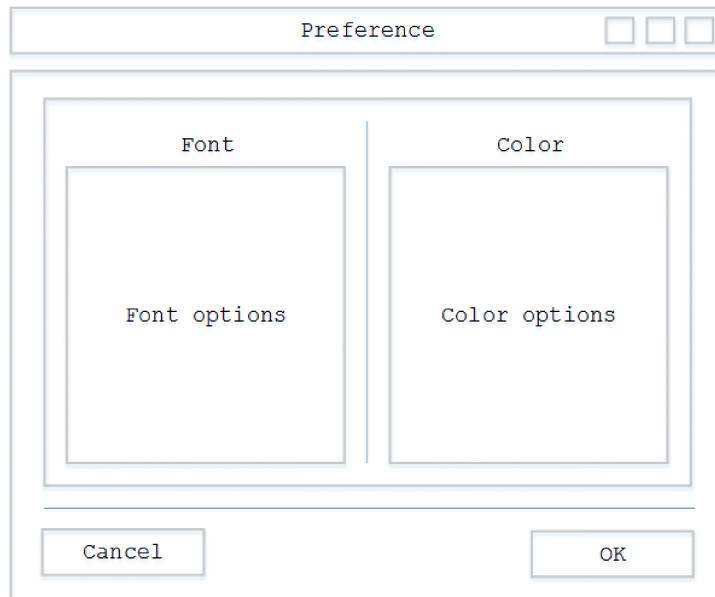
Edit Entity dialog

If a user wants to make any change in records that were previously created, he can do so by selecting the entity he wants to update, then clicking the "Edit" button. Finally, this dialog appears.



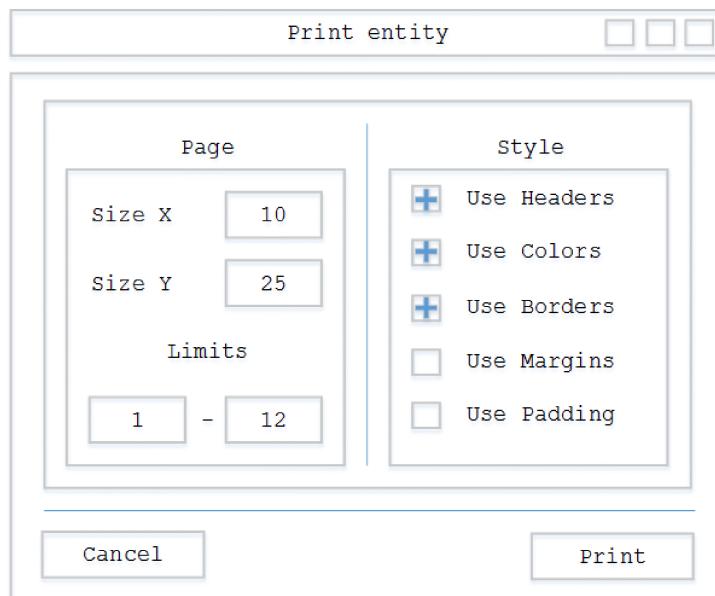
Set Mode dialog

Set Mode dialog is useful when it is necessary to assign certain rights and privileges for the user and administrator accounts. You can explicitly show what the user can and cannot do by marking corresponding checkboxes.



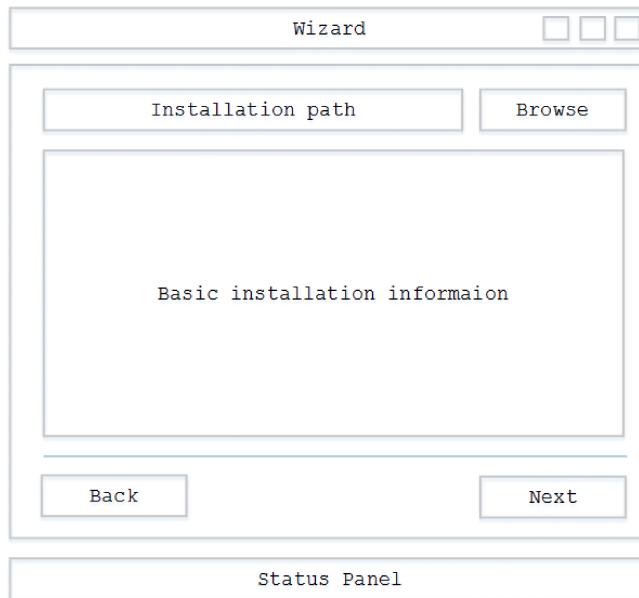
Preference dialog

Preferences help to improve user's experience when using the UTBAS application. One can adjust colors and font used in the application.



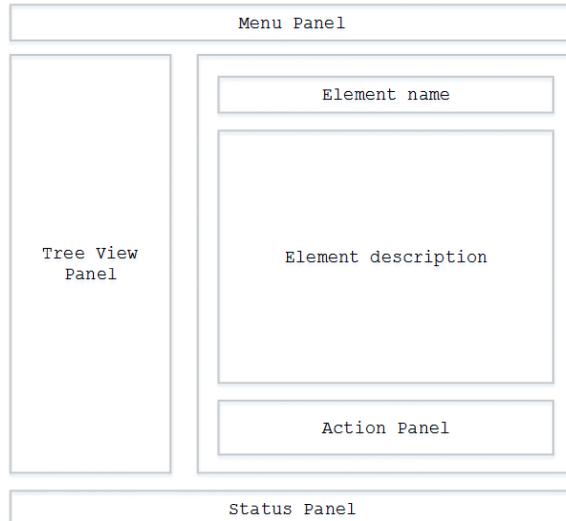
Print Entity dialog

If a user wants to create a report, he needs to select the entity on which the report will be based. Right after that he can click the "Print" button and this dialog will appear prompting the user to set the options for printing.



Installation Wizard dialog

Installation Wizard dialog helps a user to successfully complete the installation process by asking him a few questions step by step so the user can be focused on a specific question. It reduces the risk of the application to be installed incorrectly.



General interface overview

General application interface is introduced as a set of panels. The set consists of panels each of which has its own location and a specific purpose. Tree view panel provides a user with tools that help him to navigate through a set of entities and find the certain one he is searching for. It can be useful when dealing with hundreds or thousands of entities.

The second panel that is located on the right side of the main window is Description table. The Description table provides meaningful description to a selected element. The panel that locates at the bottom of the Description panel is Action panel. There is a place where all control elements will be kept.

4 Work schedule

Schedule

Task	10.26						11.2						11.9						11.16						11.23						Responsible	Completion
	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S				
SRS																													Both	100%		
SDD																													Both	100%		
GUI interface																													Andrey	15%		
PDF subsystem																													Igor	12%		
Application manager																													Both	-		
Record manager																													Igor	-		
Database schema																													Andrey	-		
Networking subsystem																													Igor	-		
Security subsystem																													Both	-		
Installation wizard																													Andrey	-		
Presentation																													Both	-		
Testing																													Both	-		
Improvements																													Both	-		

- Background task
- Repeated task
- Mutual task
- Andrey task
- Igor task