Admissions System
University of Arkansas at Little Rock

BINS 7307 Fall 2016
System Analysis and Design
Term Project
12/05/2016

Ian Calder
Simeon Ojile
Hemant Kumar
Praneeth Arjun Munagala

Table of Contents

Project Summary

Our Systems Analysis and Design Team selected an upgrade to an existing system for our term project. The project centers around enhancement of the University of Arkansas at Little Rock (UALR) existing web based Admissions System. The enhancement was brought forward by request from the Systems Manager at the University's Admission office.

The project is referred to as the Admission System, with the essence of the enhancement being modifications to address forth coming changes to funding. The Admission System itself is quite sophisticated in its workflows however the enhancement is applied only to targeted areas. With analysis of the request it was identified that additional functionality could be extended to Administrators of the System and accounted for as part of the enhancement.

The timeline for the project was defined as ‘As Soon As Possible’.

Description of the Organization and problems

Description

The University of Arkansas at Little Rock (UALR) is a metropolitan research university that provides accessibility to a quality education through flexible learning and unparalleled internship opportunities.

UALR Information Technology Services (ITS) provide the university with a framework for collaboration, enhanced awareness and increased efficiency by delivering strategically aligned IT projects at the right time and with the right resources.

ITS strives to streamline processes, increase communication, awareness and visibility into computing services to successfully manage strategically-aligned projects. It is the department’s goal to create a functional, secure, and accessible web presence.

Problem

With the forth coming changes in funding, the Universities Admissions department has requested enhancement of the existing Web based Admission System to address ‘pre-admit’ functionality. The enhancement is to provide for available program selection and, overrides of certain admissions application types with an undeclared flag. The admissions system will no longer ask the users of pre-defined admission applicant types which program they are interested in.

Project initial study

1) Objectives

The enhancement is to address the business needs to automate current manual query and reporting processes to properly categorize pre-admission applicants. Prospective student’s selection of an area of interest is to be replaced with program selection. However, not all applicants are allowed or should be in a program. For example, high school concurrent students. These special cases will no longer be allowed to choose a program and will be set as undeclared.

2) Scope definition

System Request

Project Name:	Admissions Systems
Project sponsor:	Linda Czirr
Department:	Office of Admissions
Phone:	(501) 569-8259
E-mail:	laderden@ualr.edu
Business need:	To associate students to programs accordingly

Functionality:

Using the Web, applicants complete a request for admission to UALR. The enhancement to address 'pre-admit' changes which will override the following non-degree Admissions application types with a undeclared flag.

- WD - Non-Degree
- HS - High School Concurrent
- WH - Dual enrolled
- WX - Visiting

Modify the Area of Interest Page:

- Dynamically hide/show the First and Second choice dropdowns.
- Base when to hide/show on the current application type.
- Application type override decision will be stored in a table for lookup
- If showing the dropdowns, complete application as normal.
- If hiding the dropdowns, dynamically force the area of interest as defined in the lookup table.

Build a new table to store the override definitions for when to show/hide the programs.

Build an administration page to add/remove which application types Hide/show the Area of Interest dropdowns.

The system will provide an interface for the Administrator to define time frame for degree seeking applicants.

Expected value:

Tangible:

- As an in-house initiative there are no direct development cost.
- As an enhancement there will be no cost incurred for hardware or software, all systems are existing
- Refined enrollment categorization, ensures that funding is utilized fully

Intangible:

- ITS familiarity and proficiency with the existing system
- Improved User experience with the reduction in input fields
- Improved Administrative user satisfaction.
- Improved Admissions data, for enrollment accountability, and reporting
- With the development resources and equipment at hand the enhancement can be completed and deployed with accuracy and without downtime.

3) Feasibility study

Technical Feasibility (Low Risk) Familiarity with application (High):

- The Admissions Department is well versed in their systems functionality.
- ITS is highly experienced with the Internet-based Admissions System.

Familiarity with technology (High):

- ITS built and maintains the Internet application and the entirety of the University's intranet.
- Development tools for the Web application are in use by ITS and very familiar to the team.
- The request is an enhancement of an existing system, which is solid and stable.

Project size (small):

- The project is estimated to be small in size.
- Enhancement of the system will provide for greater administrative capability, without ITS engagement.
- User interface will be marginally reduced.

Economic feasibility (excellent)

Tangible costs and benefits:

- As an in-house initiative there are no direct development cost.
- As an enhancement there will be no cost incurred for hardware or software, all systems are existing
- Refined enrollment categorization, ensures that standing for funding is utilized fully

Intangible costs and benefits:

- ITS familiarity and proficiency with the existing system
- Improved User experience with the reduction in input fields
- Improved Administrative user satisfaction.
- Improved Admissions data, for enrollment accountability, and reporting
- With the development resources and equipment at hand the enhancement can be completed and deployed with accuracy and without downtime.

Organizational feasibility (excellent):

Project champion:

- Systems Manager, Office of Admissions

Executive management:

- There is strong support of the project within the executive management team.

Users:

- The end users are the driving force for the Universities educational services

Other stakeholders:

- ITS is making every effort to do their part increase enrollment through the means of services and support in a technological manner are being prioritized.

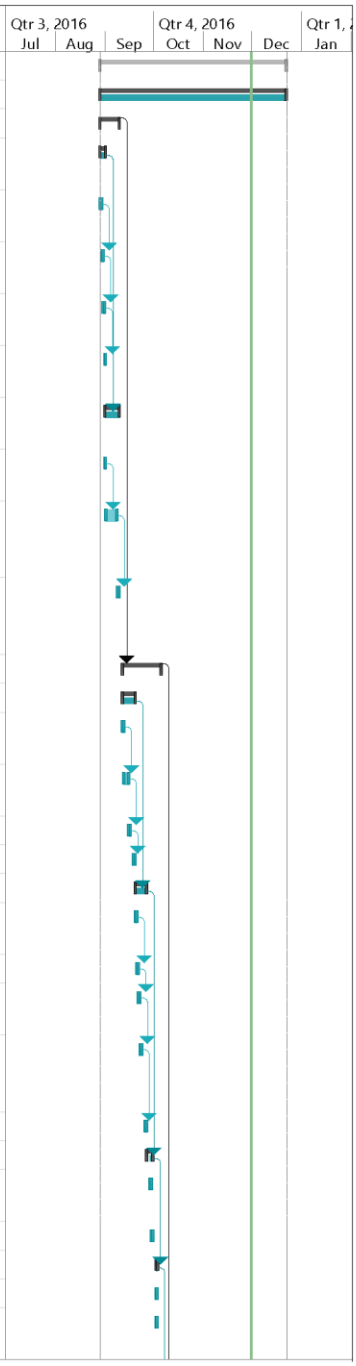
Additional comments:

- The University views this as a strategic system. It is the first point of entry for prospective students.
- Primary driver for the University is enrollment.
- The end user's experience should be a paramount concern with any system across the University but especially at the initial point of engagement.

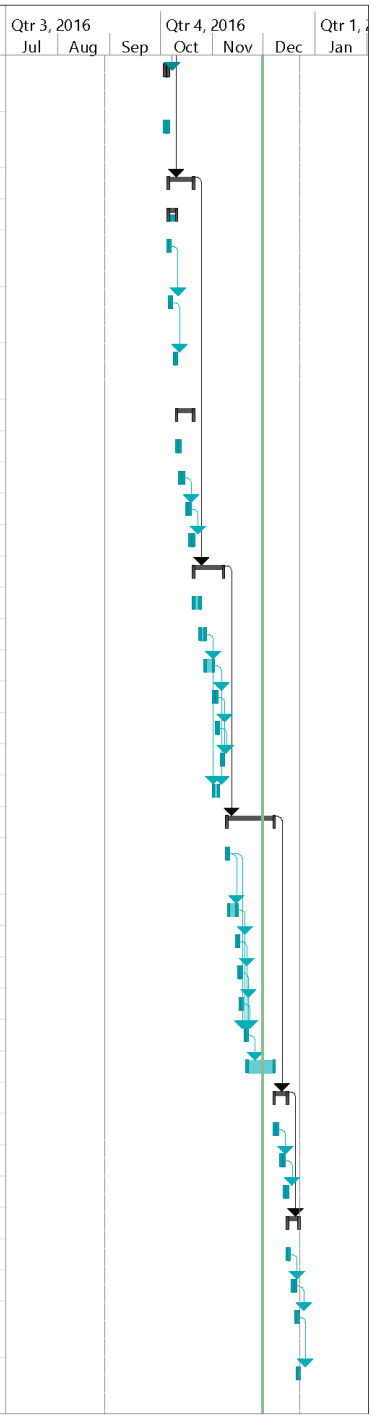
Project Management Plan

1) Work plan (Gantt)

ID	Task Name	Start	Finish	Duration
0	**SAD Project Work-Modified**	**8/29/2016**	**12/22/2016**	**84 d**
1	**UALR Admissions System**	**8/29/2016**	**12/22/2016**	**84 d**
2	**Initiating**	**8/29/2016**	**9/9/2016**	**10 d**
3	**Develop Project Initial Study**	**8/29/2016**	**9/1/2016**	**3.5 d**
4	Identify Goals and Objectives	8/29/2016	8/29/2016	1 d
5	Develop Strategies and Plans	8/30/2016	8/30/2016	1 d
6	Research Existing System	8/31/2016	8/31/2016	1 d
7	Develop Feasibility Study	9/1/2016	9/1/2016	0.5 d
8	**Develop Preliminary Project Scope Statement**	**9/1/2016**	**9/9/2016**	**7 d**
9	Document Project Costs and Benefits	9/1/2016	9/1/2016	0.5 d
10	Prepare Preliminary Project Scope Statement	9/1/2016	9/8/2016	5 d
11	Develop High Level Work Breakdown Structure	9/9/2016	9/9/2016	1 d
12	**Planning**	**9/12/2016**	**10/5/2016**	**18 d**
13	**Define Scope**	**9/12/2016**	**9/19/2016**	**6 d**
14	Document Scope Management Plan	9/12/2016	9/12/2016	1 d
15	Specify Deliverables and Acceptance Criteria	9/13/2016	9/15/2016	3 d
16	Define Scope	9/16/2016	9/16/2016	1 d
17	Document Assumptions	9/19/2016	9/19/2016	1 d
18	**Develop Project Schedule**	**9/20/2016**	**9/26/2016**	**5 d**
19	Build Work Breakdown Structure	9/20/2016	9/20/2016	1 d
20	Develop Resource Plans	9/21/2016	9/21/2016	1 d
21	Prepare Project Estimates	9/22/2016	9/22/2016	1 d
22	Define Dependencies and Develop Project Schedule	9/23/2016	9/23/2016	1 d
23	Document Assumptions	9/26/2016	9/26/2016	1 d
24	**Organize Project Resources**	**9/27/2016**	**9/30/2016**	**4 d**
25	Develop Organization Structure	9/29/2016	9/29/2016	1 d
26	Develop Staffing Plan	9/30/2016	9/30/2016	1 d
27	**Assign Project Team**	**10/3/2016**	**10/3/2016**	**1 d**
28	Define Resources	10/3/2016	10/3/2016	0.5 d
29	Conduct System Orientation	10/3/2016	10/3/2016	0.5 d



ID	Task Name	Start	Finish	Duration
30	**Develop Project Management Plan**	**10/4/2016**	**10/5/2016**	**2 d**
31	Document Project Management Plan	10/4/2016	10/5/2016	2 d
32	**Analysis**	**10/6/2016**	**10/20/2016**	**11 d**
33	**Requirement Analysis**	**10/6/2016**	**10/10/2016**	**3 d**
34	Identify User Requirements	10/6/2016	10/6/2016	1 d
35	Prioritize User Requirements	10/7/2016	10/7/2016	1 d
36	Finalise User Requirements	10/10/2016	10/10/2016	1 d
37	**System Analysis**	**10/11/2016**	**10/20/2016**	**8 d**
38	Activity Diagram	10/11/2016	10/12/2016	2 d
39	Use Case Diagram	10/13/2016	10/14/2016	2 d
40	Class Diagram	10/17/2016	10/18/2016	2 d
41	Sequence Diagram	10/19/2016	10/20/2016	2 d
42	**System Design**	**10/21/2016**	**11/7/2016**	**12 d**
43	Application Architecture	10/21/2016	10/24/2016	2 d
44	Class Design	10/25/2016	10/27/2016	3 d
45	Object Persistence Design	10/28/2016	11/1/2016	3 d
46	User Interface	11/2/2016	11/3/2016	2 d
47	Input	11/4/2016	11/4/2016	1 d
48	Output	11/7/2016	11/7/2016	1 d
49	Program Logic Specification	11/2/2016	11/4/2016	3 d
50	**Coding and Implementation**	**11/10/2016**	**12/7/2016**	**20 d**
51	Configure and Set Up Application Architecture	11/10/2016	11/10/2016	1 d
52	Relational Database	11/11/2016	11/15/2016	3 d
53	User Interface	11/16/2016	11/16/2016	1 d
54	Input	11/17/2016	11/17/2016	1 d
55	Output	11/18/2016	11/18/2016	1 d
56	Program Logic Specification	11/21/2016	11/21/2016	1 d
57	Coding	11/22/2016	12/7/2016	12 d
58	**Testing**	**12/8/2016**	**12/15/2016**	**6 d**
59	Test Planning	12/8/2016	12/9/2016	2 d
60	System Testing	12/12/2016	12/13/2016	2 d
61	User Acceptance Testing	12/14/2016	12/15/2016	2 d
62	**Conversion**	**12/16/2016**	**12/22/2016**	**5 d**
63	Go-Live	12/16/2016	12/16/2016	1 d
64	Assess Satisfaction	12/19/2016	12/20/2016	2 d
65	Summarize Project Results and Lessons Learned	12/21/2016	12/21/2016	1 d
66	Close Out the Project Records	12/22/2016	12/22/2016	1 d



2) Size, duration, cost estimation

Function Point Calculation

Description	Complexity			Total
	Low	Medium	High	
Inputs	2 x 3	_____ x 4	_____ x 6	9
Outputs	1x 4	_____ x 5	_____ x 7	4
Queries	3 x 3	_____ x 4	_____ x 6	9
Files	0 x 7	_____ x 10	_____ x 15	0
Interfaces	2 x 5	_____ x 7	_____ x 10	10

Total Unadjusted Function Points (TUFP): ____***29***____

The system enhancement is not sophisticated

Adjusted project complexity (PCA)
PCA = 0.65 + (0.01 x PC)

Short Cut

i) Simple system: 0.65
ii) Normal system: 1
iii) Complex system: 1.35

PCA= 0.65 + (0.01 * 0.65)
PCA= 0.66

Total adjusted function points (TAFP)

TAFP = TUFP x PCA
TAFP = 29 * 0.66
TAFP = 19.14

Approximate Number of Lines of Code per Function Point

Language	Lines of code per FP
PHP	67
HTML	40
JavaScript	53

The Project would use a combination of PHP and JavaScript (for most programs) and HTML for the web screens. 50% of the function points would be PHP, 20% would be HTML and 30% in JavaScript.

= 19.14 * 0.50 * 67 + 19.14 * 0.20 * 40 + 19.14 * 0.3 * 53
= 641.19 + 153.12 + 304.33
= 1098.64 LOC
= 1.1 KLOC

Project Duration

- MM = $a(KLOC)^b$ x EAF
- D = $c(MM)^d$
- EAF for the basic model is 1

SW Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

(Basic Model)

Function Point = 1.1 KLOC project

Organic mode:

- Effort = 2.4 $(1.1)^{1.05}$ = 2.65 MM
- Schedule = 2.5 $(2.65)^{0.38}$ = 3.6 months
- MM = $10,000
- Average Staffing = 0.74
- Cost:0.74 * $10,000 = $7,400.00

3) Staffing and Team Organization

Team:

- Brian Keltch, Director Enterprise Applications & Development
- Donald O'Donnell, Project Manager
- Daniel Spillers, Lead Web Developer
- John Franks, Application Developer
- Robert Brown, Database Administrator
- Curt Thompson, Release and Deployment Manager

User Requirements Analysis

1. Non degree seeking applicants will be automatically defined as non-declared
    1.1 The system will not display area of interest fields for non-degree seeking applicants

    1.2 The system will provide a configuration screen for administrators to define overrides for Application types

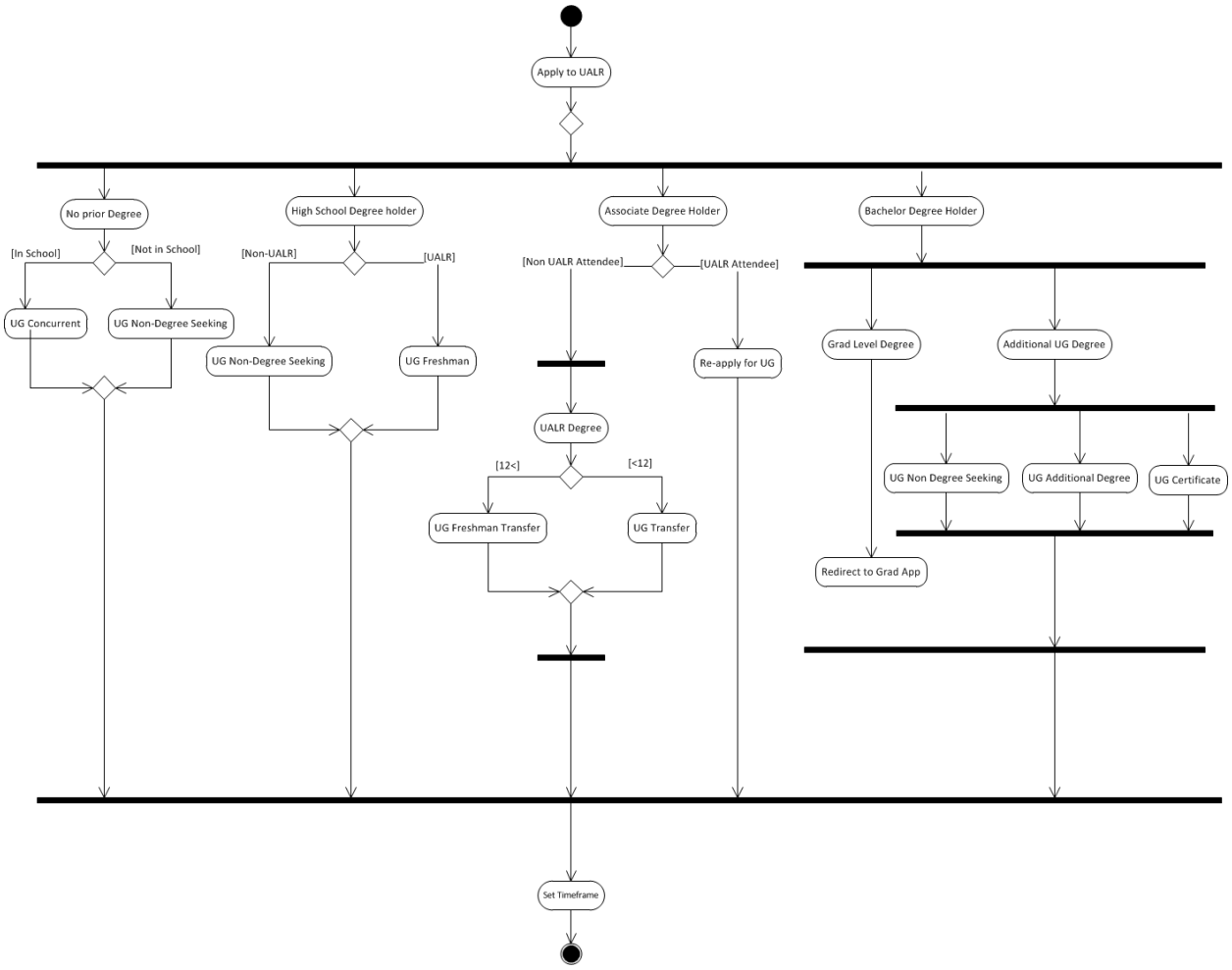
2. For degree seeking applicants the system will provide an administrative interface
    2.1 The system shall display area of interest fields for degree seeking applicants

    2.2 The system will allow Administrators to define time frame for applicants to declare their major

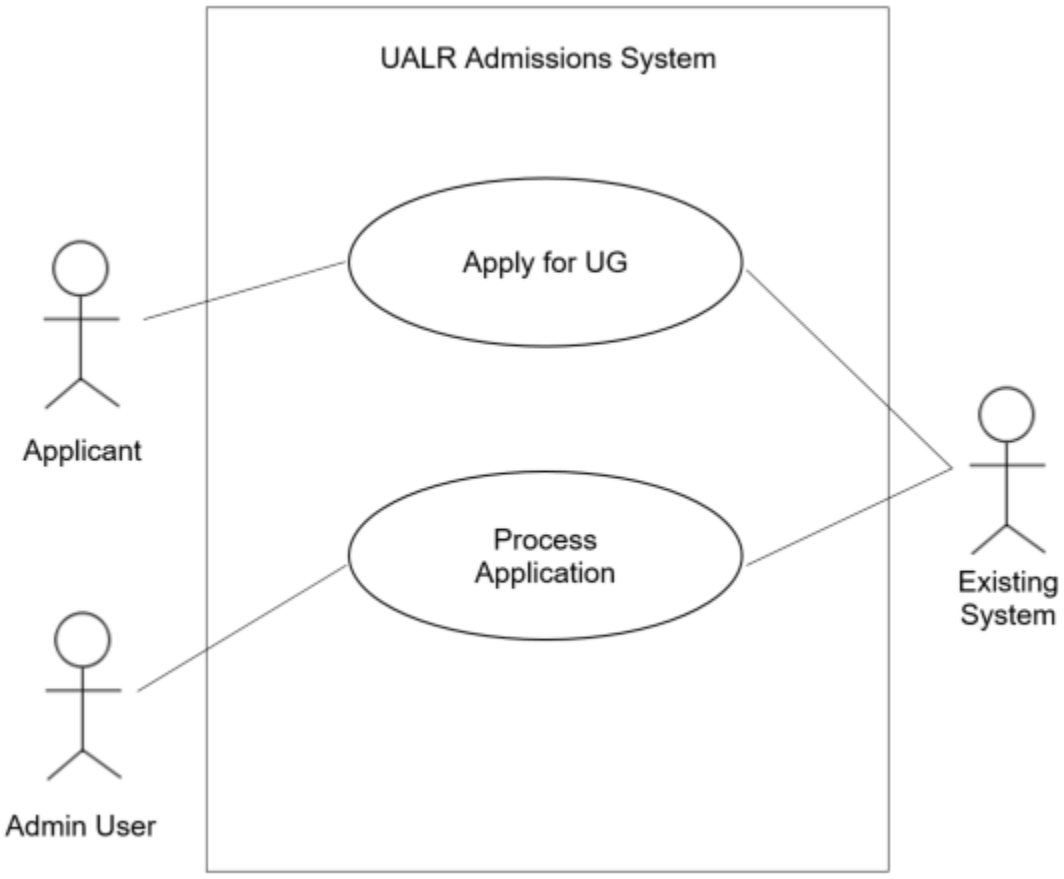
    2.3 The system shall declare the degree seeking applicant’s area of interest as their major after the completion of the defined time period.

System Analysis and design

1) Activity Diagram



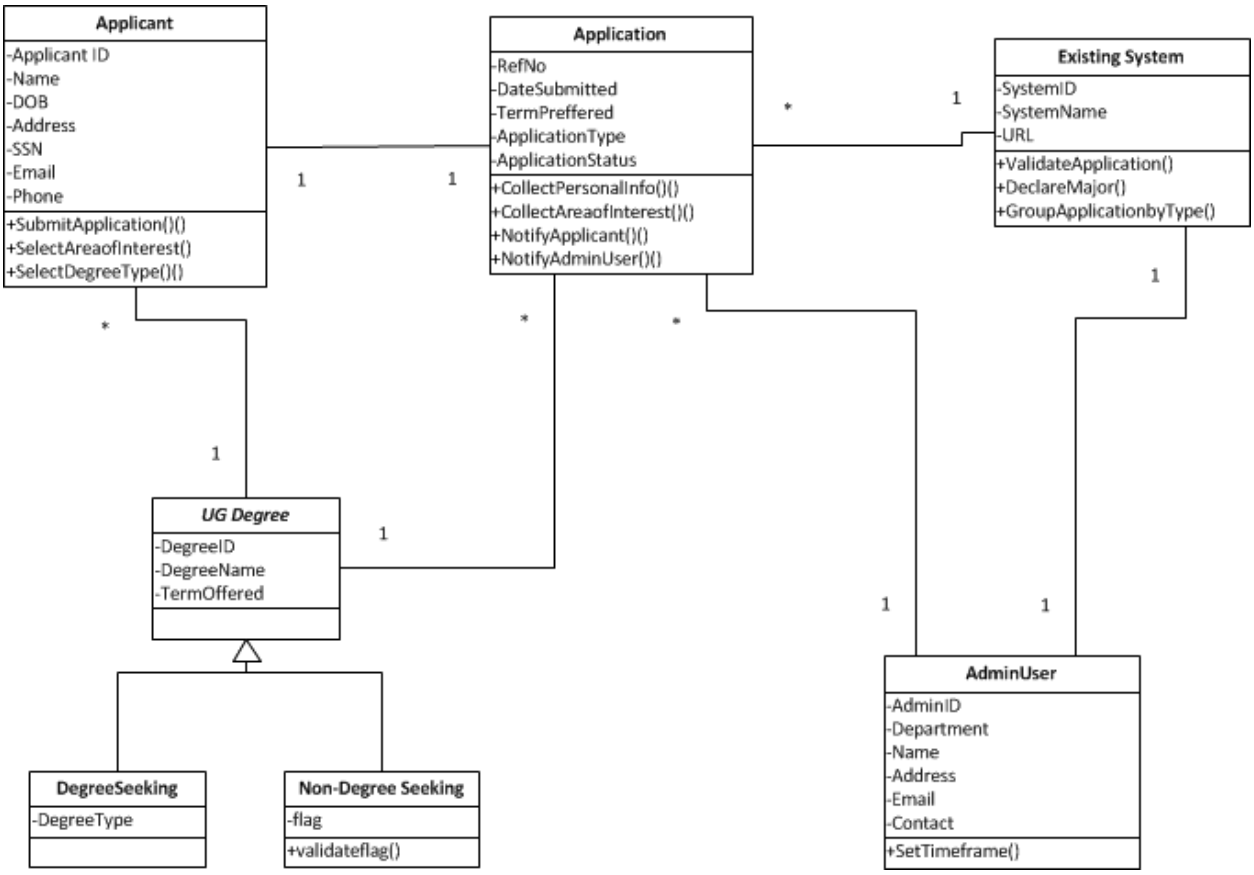
2) Use Case Diagram and Use Case Description



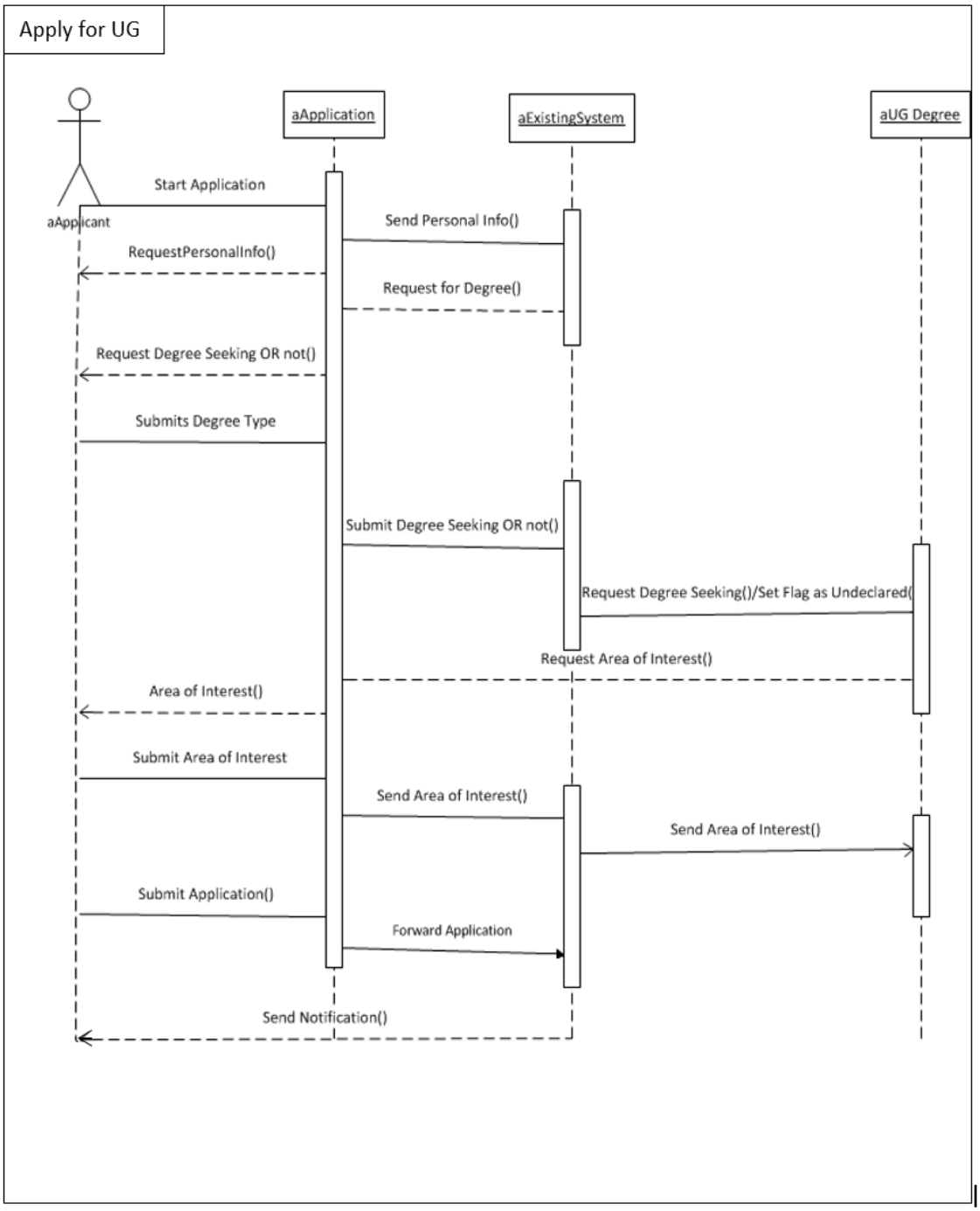
Use Case Name	Apply for UG
Actors	Applicant, Existing System
Description	This use case describes the process of an Applicant submitting a new application for an UG degree. On completion, the Applicant will receive a notification with a unique reference number.
Basic Course	Step1. This use case is initiated when an Applicant submits an application to be processed. Step 2. The applicant's partial personal information is requested by the system. Step 3. The system will request for a degree seeking or a non-degree seeking option. Step 4. If an applicant chooses degree seeking option, an area of interest should be specified. Step 5. The applicant's complete information is requested by the system. Step 6. The applicant will be prompted to submit the application. Step 7. Upon submission, the system will generate an email notification with a reference number for further inquiry.
Alternative Courses	Step 1. If the Applicant is a returning user, the applicant's personal information is validated against what is currently on file Step 4. If an applicant chooses non-degree seeking option. 1. The application is flagged as undeclared..
Others	Any comments

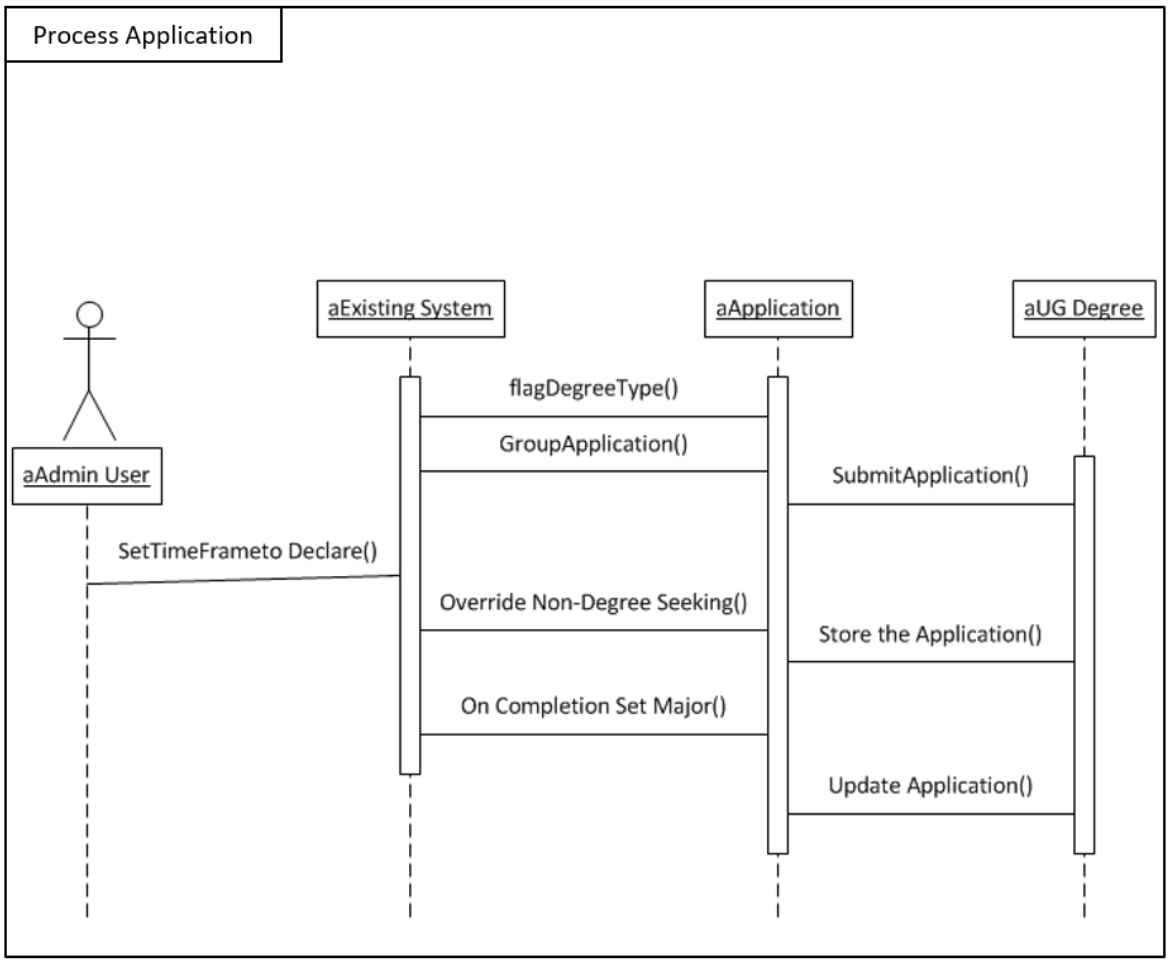
Use Case Name	Process Application
Actors	Existing System, Admin User
Description	This use case describes the processing of a standard UG application by the existing system and the Admin user. A notification with the admission status will be sent to the specified address.
Basic Course	Step1. This use case is initiated when a standard UG application is submitted. Step 2. The application on file is validated and reviewed. Step 3. For all applications, the existing system will flag records with degree type. Step 4. The admin user will allocate a set timeframe for degree seeking applications to declare their chosen Major. Step 5. If the application is undeclared upon completion of the set timeframe, the system will automatically declare the chosen area of interest as the applicant's major. Step 6. The applicant will be accordingly notified of the system's action.
Alternative Courses	Step 2. Upon any inconsistency with the application, the applicant is notified to rectify the same. Step 4. For non-degree seeking applications, the existing system will flag records as undeclared.
Others	Any comments

Class Diagram

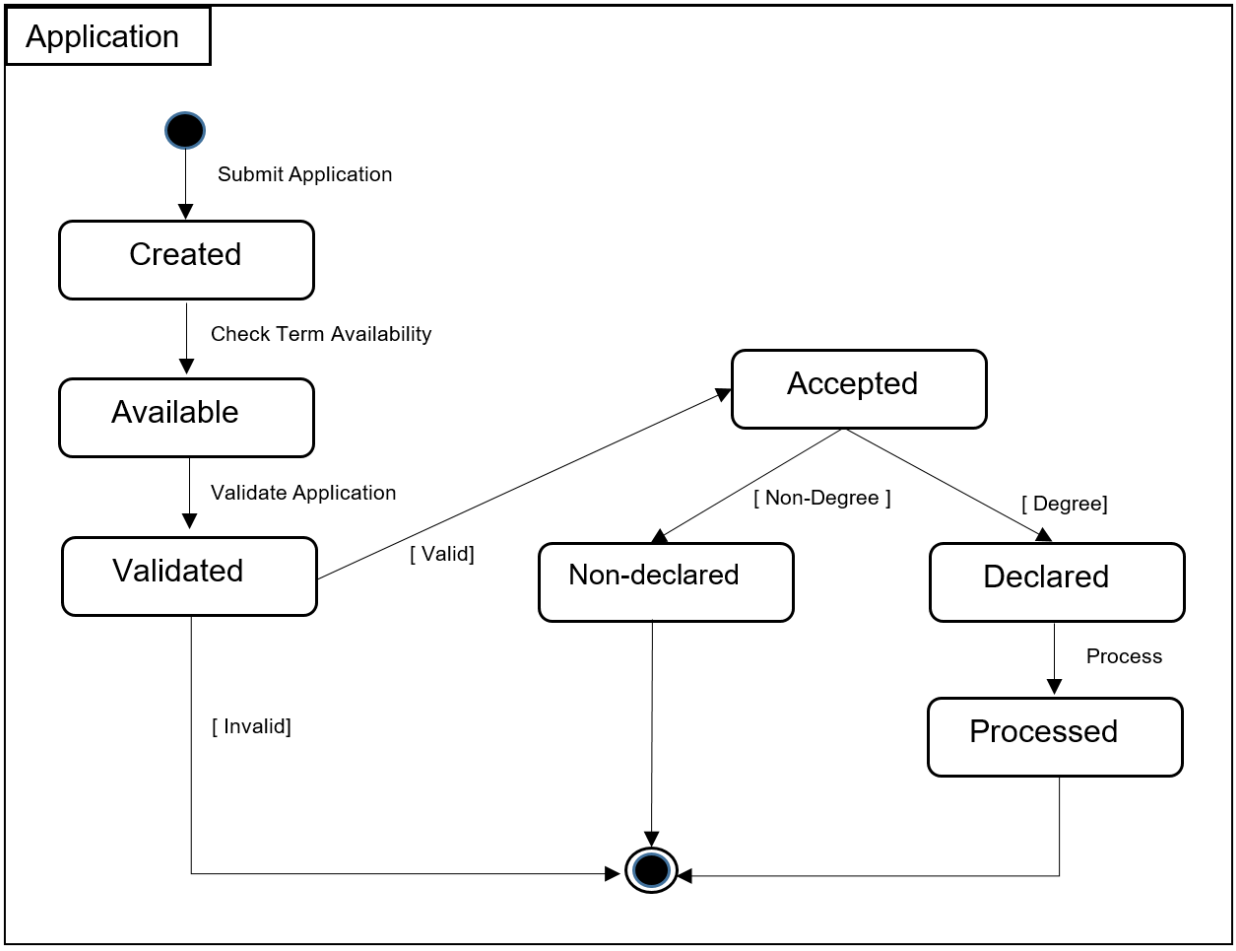


Sequence Diagrams




Process Application
aExisting System
aApplication
aUG Degree
aAdmin User
flagDegreeType()
GroupApplication()
SubmitApplication()
SetTimeFrameto Declare()
Override Non-Degree Seeking()
Store the Application()
On Completion Set Major()
Update Application()

State Machine Diagram



Object Persistence Design

EXISTING_SYSTEM

PK

SystemID	System_Name	URL

UG_DEGREE

PK

DegreeID	Degree_Name	Term_Offered

DEGREE_SEEKING

PK/FK: UG_DEGREE.DegreeID

DegreeID	Type	Length

NON_DEGREE_SEEKING

PK/FK: UG_DEGREE.DegreeID

DegreeID	Flag

APPLICANT

PK FK: UG_DEGREE.DegreeID

ApplicantID	Name	DOB	SSN	Phone	Address	Email	DegreeID

ADMIN_USER

PK FK: EXISTING_SYSTEM.SystemID

AdminID	Department	Name	Address	Email	Contact	SystemID

APPLICATION

PK FK: EXISTING_SYSTEM.SystemID FK:UG_DEGREE.DegreeID

RefID	DateSub	TermPref	AppType	AppStatus	ApplicantID	SystemID	AdminID	DegreeID

FK: APPLICANT.ApplicantID FK: ADMIN_USER.AdminID

Conclusion

With forth coming changes to the funding model it is imperative that UALRs Admissions System identify and classify students accordingly. The systems enhancement will provide for additional ease of use for the students, a greater level of self-service configuration capability for the Admission staff; providing user empowerment and system flexibility by design versus hard coding.

The end result will bring better refined data as the university continues to evolve. The level of effort is quite manageable and demonstrates the value of having in-house resources who can respond to such enhancement requests; leveraging their experience and system knowledge to provide for such timely and cost effective resolutions.

Overall benefit of the enhancement will extend directly to prospective students/ users, Admissions Staff and indirectly to ITS and the University of Arkansas at Little Rock as a whole; from the perspective of enrollment.