Table of Contents

1. Introduction

1.1 Purpose of Document

This requirement documentation was created to represent the new “Blueprint” Study Plan System (BSPS) for Illinois College. The “Blueprint” was imposed by Illinois College in order to allow students to acquire a complete Liberal Arts education. The “Blueprint” contains different academic course categories that a student must complete to graduate from Illinois College. The purpose of the system is to enhance the understanding of a student’s “Blueprint” progress and to alleviate the problems in the process of creating a student’s future schedule. This document describes the overview and the intended goal for the new system. This document models the non-functional, referring to the properties of the system, and functional requirements described with use cases, interaction diagrams, and class models. System design and implementation should be directed from this documentation.

1.2 Project Summary

Project Name:	Development of a Generic Educational Advising framework for Facilitating students, faculty, and staff to coordinate study plans
Project Manager:	Madison Niederer
Project Analysts:	Son Diep, Analyst Ye Htet, Analyst Jonah Pattie, Analyst Omar Urquiza, Analyst
Responsible Users:	Illinois College Students Illinois College Faculty IC Staff

1.3 Background

Going into college or university can be daunting, so being able to set up a rough framework of what a student will need to accomplish at school is important. It can also be the student's first impression of that particular school's infrastructure if the interface is difficult to manage, or there are many misunderstandings then it can be frustrating. Having a simple program that is clean and intuitive to use would be the most beneficial for students, advisors, and staff. As for Illinois College's specific requirements for graduation, i.e the "Blueprint", we will add features that will improve the usability and remove those design elements that create unnecessary difficulty throughout the advising process. These include difficulties with: auto-generated advising sheets, course prerequisites, course times and dates, course credits and blueprint credits. Problems within the UI include: poor navigation, confusing directions, and confusing descriptions.

1.4 Project Scope

The scope of this project is a system framework that can serve as the skeleton for any advising system for higher educational institutions. As such, the individual implementation of the product will be relatively local and small in scope. However the generalized forms and logic should be "plug and play" applicable to many or all institutions. As such, the scope of the project in design is global, however the individual implementations will be local.

1.5 System Purpose

1.5.1 System Users

Staff

- **Registrar**
  - A Registrar Staff is someone who has the control to edit and update all the course data. This includes adding new courses, deleting courses, editing the description of the courses and changing the requirements the courses fulfill. Through the BPSP, a registrar staff can manipulate all these data and information pertaining to courses.
- **IT**
  - IT staff will be in charge of maintaining the BSPS. They will have access to the inner design of the system and be able to add new features or fix the system.

Faculty

- **Advisor**
  - An Advisor is someone who advises a student on planning their college career. They can check the courses the students have chosen and confirm the choices.

They also have access to the student recommendation list and other credits of their student.

Students

- **Student**
    - Students are the primary users of the system. They seek to register courses and plan on how they want their college career to look like. BSPS provides a recommendation list and an advising sheet that can help during advising sessions.

1.5.2 Location

The application will be used in an online format using either a computer system, or mobile system. This application will be available to anyone with an email account within the institution's domain. That is, if a user has the credentials to access the institution's version of this application they will be able to do so wherever they are given they have internet access, and a device able to run a browser.

1.5.3 Responsibilities

The primary responsibilities of the new system include:

- Create simple, user-friendly, graphical way to find and understand courses
    - Have clear course descriptions that layout the courses meeting location, times, credits, and blueprint / secondary credits. For a generalized application, this could be the institution's form of 'reading extensive' or 'social, spiritual, philosophical' etc.
    - Have easy to use dialogue that allow for the searching of courses by various criteria, an example of this would be: "show me courses that meet at 8:00 am and fulfill an ethical credit"
- Be able to provide recommended courses for a student
- Be able to generate an advising report for a student
- Be applicable in many institutional settings
    - Generalized forms and logic should allow for easy plug and play of our system to fit within any institutional setting.
- Be usable in a variety of formats
    - System should be usable for most devices
        - Computer Systems
            - Windows OS
            - Apple OS
            - Linux OS
        - Mobile Systems
            - Android

- Apple

1.5.4 Application Needs

The new framework is needed to:

- Function in an online, electronic environment
- Have access to the institutions course Database
- Have access to the institutions user/account information
    - User ID's and transcripts
    - This must be secure as such must require user password/passcode
- Function across many platforms (listed in 1.5.3)
- Allow a Superuser, i.e. an advisor to access an advisee's transcript information
- Allow printing across a LAN or wireless network should a user wish to print their information.

1.6 Overview of Document

The rest of this document gives the detailed specifications for the new educational advising system. It is organized as follows:

- Section 2: Functional Objectives
  Each objective gives a desired behavior for the system, a business justification, and a measure to determine if the final system has successfully met the objective. These objectives are organized by priority. In order for the new system to be considered successful, all high priority objectives must be met.
- Section 3: Non-Functional Objectives
  This section is organized by category. Each objective specifies a technical requirement or constraint on the overall characteristics of the system. Each objective is measurable.
- Section 4: Context Model
  This section gives a text description of the goal of the system, and a pictorial description of the scope of the system in a context diagram. Those entities outside the system that interact with the system are described.
- Section 5: Use Case Model
  The specific behavioral requirements of the system are detailed in a series of use cases. Each use case accomplishes a system task and shows the interaction between the system and some outside actor. Each use case is described with both text and an interaction diagram. The system use case diagram depicts the interactions between all use cases and system actors.

2. Functional Objectives

2.1 High Priority

1. Students:
    a. Generate a Recommendation Report for student. Recommendation Report lists possible course selections for next semester that fulfil missing Blueprint and Major requirements
    b. View Transcript
2. Advisor:
    a. Generate a Student Transcript to show student's graduation date, convocation credits, and current GPA , completed courses, and Blueprint fulfilments for advisor to see if student is on track for graduation
    b. Generate an Advising Report for Advisor to check student's Blueprint and Major requirement fulfilments and course offerings next semester to fulfil each missing requirement
    c. View Master Course Schedule
3. Registrar:
    a. Allow the Registrar to update course's blueprint fulfilments, course number, course description, and course schedule for each semester.
    b. Generate Master Course Schedule for next semester using courses and course schedules added by Registrar

2.2 Medium Priority

1. Student:
    a. Offer Recommendation Report for next n-semester. Offers Recommendation Report explained previously, but for an additional n-semesters until student's graduation. Allowing them to create a 4-year plan.
    b. Alert students of course fees and other course changes when offering as a fulfillment on Recommendation Report
2. Registrar:
    a. Convert Transfer Credits to IC course fulfillments on the Blueprint and Major section.

2.3 Low Priority

1. Students and Advisor:
    a. Check Requirements for programs like Phi Beta Kappa and offer courses to fulfil in Recommendations Report, if a student has a GPA over 3.7
    b. Save previous reports to reduce redundancy and ensure access if the database is down and won't send errors when calling for reports.

3. Non-Functional Objectives

3.1 Reliability

- The system shall be completely operational at least x% of the time.
- Downtime after a failure shall not exceed x minutes.
- System should be able to withstand wrong user input.
- System should support multiple users.

3.2 Usability

- Students and advisors should be able to use the system without any previous training.
- Easy interfaces within the system to allow users to find their desired product.
- Offer flexibility for users to select different operations of the system.

3.3 Performance

- System should perform one function at a time for x users.
- System should have x response time
- Throughput is measured by how many times the user will access the system

3.4 Security

- Transaction data must be transmitted in encrypted form
- Access only available for registered users
- Redact student personal information in paper format of all reports

3.5 Supportability

- The system should be able to accommodate new products without major reengineering.
- The system should be compatible with Windows 7, 8, and 10 as well as Mac OS.

3.6 Online user Documentation and Help

- Help documentation will be implemented directly into the program

3.7 Purchased Components

- No purchases were necessary for the creation of this software

3.8 Interfaces

The system must interface with

- The current Oracle database systems to store student and class information

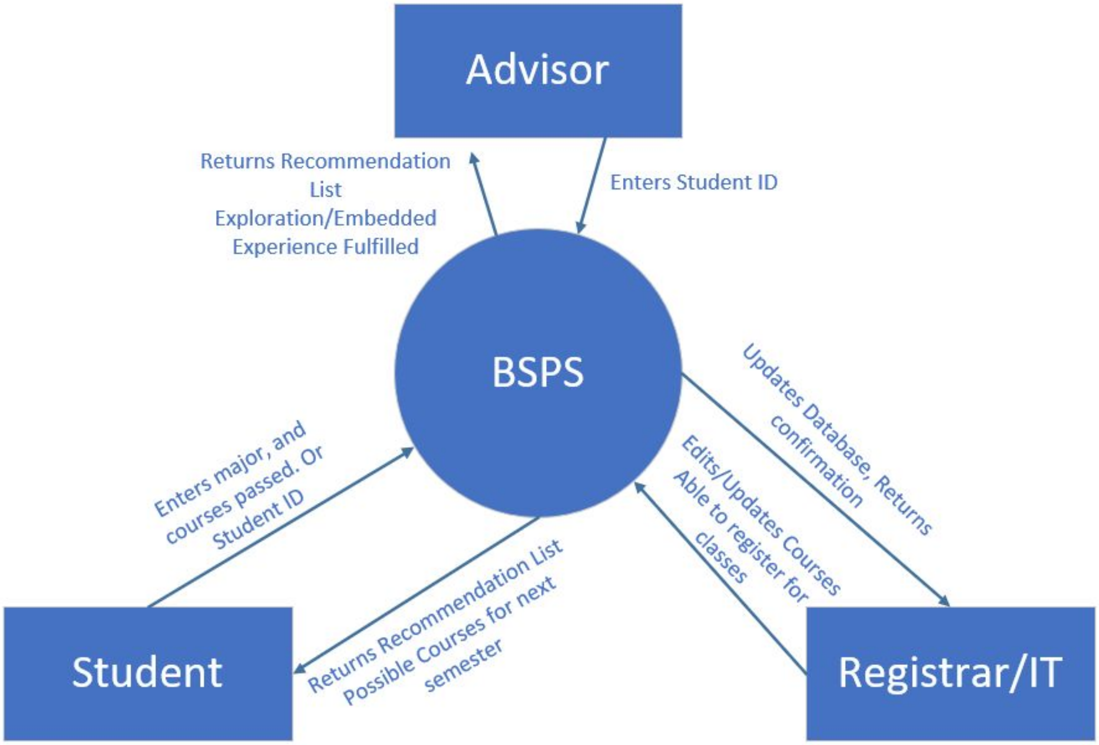
4. The Context Model

4.1 Goal Statement

- The purpose of the system is to enhance the understanding of a student's "Blueprint" progress, and to alleviate the problems in the process of creating a student's future schedule.
- Add something about making the previous system more accessible to users

4.2 Context Diagram

BSPS


Advisor
Returns Recommendation List
Exploration/Embedded Experience Fulfilled
Enters Student ID
BSPS
Updates Database, Returns confirmation
Edits/Updates Courses
Able to register for classes
Enters major, and courses passed. Or Student ID
Returns Recommendation List
Possible Courses for next semester
Student
Registrar/IT

4.3 System Externals

Staff

- **Registrar :** A Registrar staff is someone who has the control to edit and update all the course data. This includes adding new courses, deleting courses, editing the description of the courses and changing the requirements the courses fulfill. Through the BPSP, a registrar staff can manipulate all these data and information pertaining to courses.

- **IT :** IT staff will be in charge of maintaining the BSPS. They will have access to the inner design of the system and be able to add new features or fix the system.

Faculty

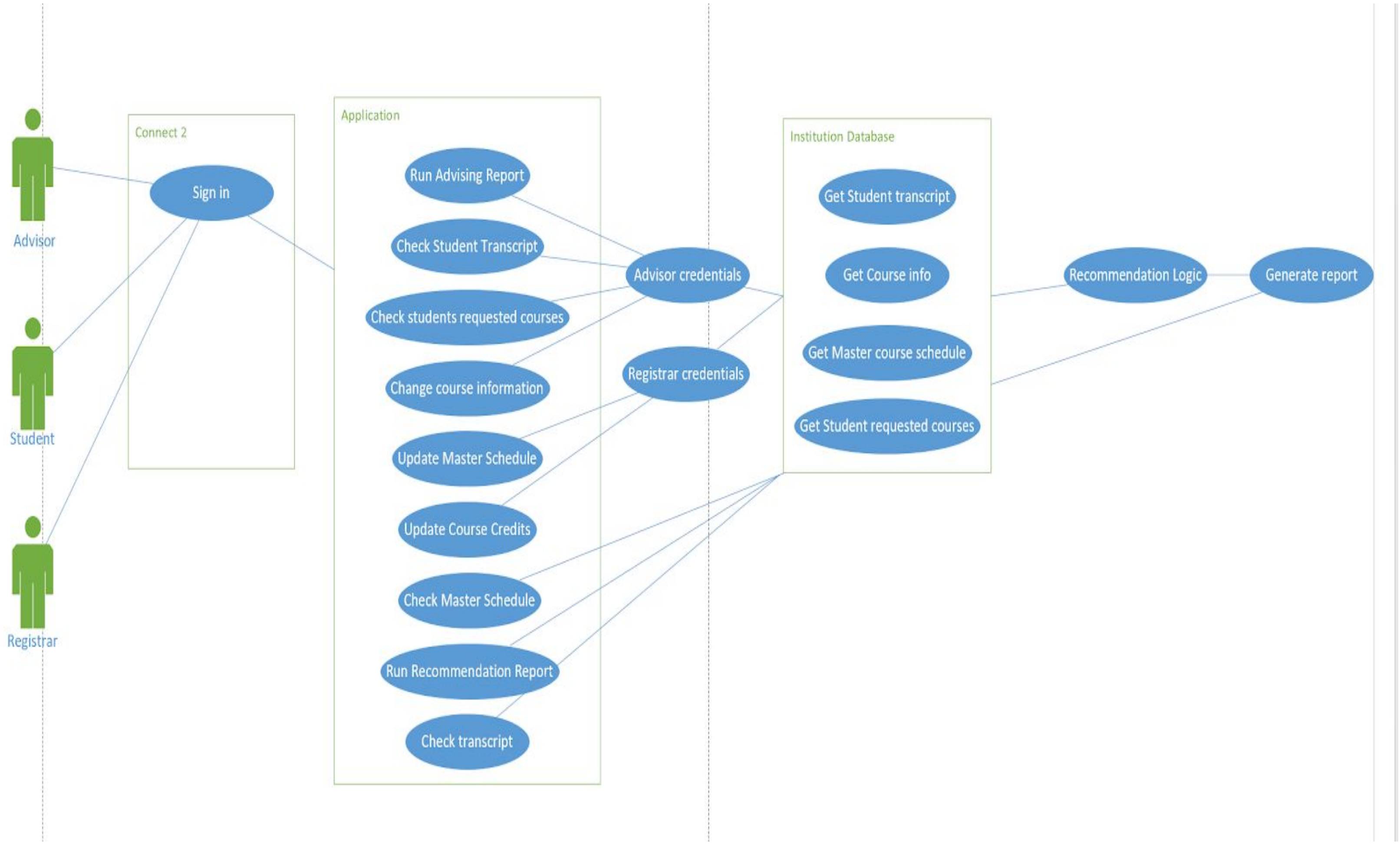
- **Advisor:** An Advisor is someone who advises a student on planning their college career. They can check the courses the students have chosen and confirm the choices. They also have access to the student recommendation list and other credits of their student.

Students

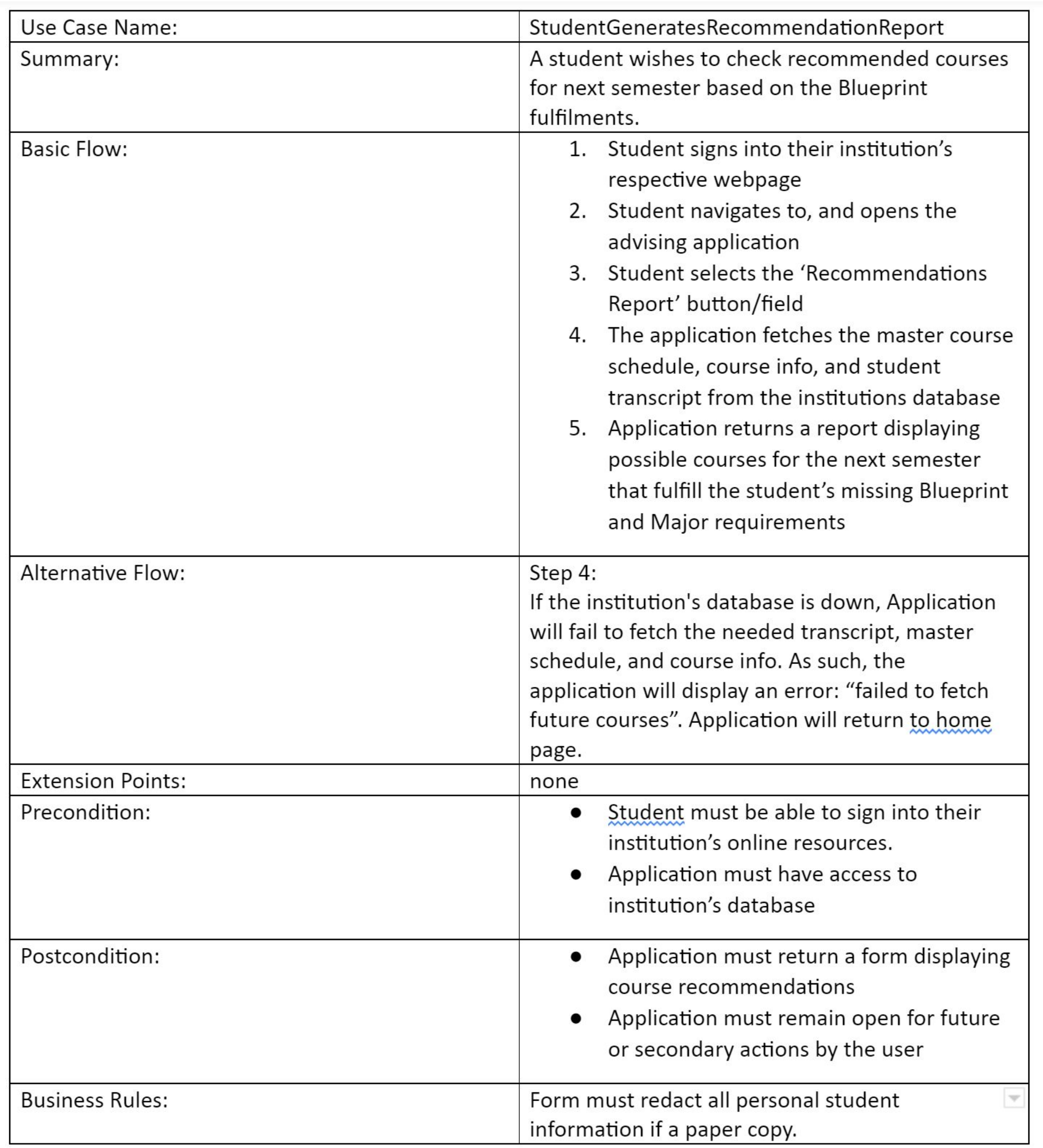
- **Student:** Students are the primary users of the system. They seek to register courses and plan on how they want their college career to look like. BSPS provides a recommendation list and an advising sheet that can help during advising sessions.

5. The Use Case Model

5.1 System Use Case Diagram



5.2 Use Case Descriptions (for selected cases)



Use Case Name:	StudentGeneratesRecommendationReport
Summary:	A student wishes to check recommended courses for next semester based on the Blueprint fulfilments.
Basic Flow:	1. Student signs into their institution’s respective webpage 2. Student navigates to, and opens the advising application 3. Student selects the ‘Recommendations Report’ button/field 4. The application fetches the master course schedule, course info, and student transcript from the institutions database 5. Application returns a report displaying possible courses for the next semester that fulfill the student’s missing Blueprint and Major requirements
Alternative Flow:	Step 4: If the institution's database is down, Application will fail to fetch the needed transcript, master schedule, and course info. As such, the application will display an error: “failed to fetch future courses”. Application will return to home page.
Extension Points:	none
Precondition:	● Student must be able to sign into their institution’s online resources. ● Application must have access to institution’s database
Postcondition:	● Application must return a form displaying course recommendations ● Application must remain open for future or secondary actions by the user
Business Rules:	Form must redact all personal student information if a paper copy.

Use Case Name:	AdivsorChecksFutureCourses
Summary:	An Advisor wishes to check the future courses offered by their respective institution
Basic Flow:	1. Advisor signs into their institution's respective webpage 2. Advisor navigates to, and opens the advising application 3. Advisor selects the 'Check future courses' button/field 4. The application fetches the master course schedule from the institutions database 5. Application returns a window displaying the courses offered for future semesters
Alternative Flow:	Step 4: If institutions database is down, Application will fail to fetch master course schedule. As such, the application will display an error: "failed to fetch future courses". Application will return to home page.
Extension Points:	none
Precondition:	• Advisor must be able to sign into their institution's online resources. • Application must have access to institution's database
Postcondition:	• Application must return a form displaying the future available courses • Application must remain open for future or secondary actions by the user
Business Rules:	none

Use Case Name:	RegistrarUpdatesMasterCourseSchedule
Summary:	A Registrar staff member manipulates the master course schedule, usually the institution’s database.
Basic Flow:	1. Registrar staff signs into the institution’s respective webpage 2. Registrar staff navigates to, and opens the master course schedule section 3. The application fetches the master course schedule from the institutions database 4. Registrar staff search for course to update 5. Registrar staff make changes to the course, such as changing course description or credit.
Alternative Flow:	Step 3: If institutions database is down, Application will fail to fetch master course schedule. As such, the application will display an error: “failed to fetch future courses”. Application will return to home page.
Extension Points:	none
Precondition:	• Registrar staff must be able to sign into their institution’s online resources. • Application must have access to institution’s database
Postcondition:	• Application must return a form displaying the future available courses • Application must remain open for future or secondary actions by the user
Business Rules:	none