American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Fall 19_20

Section: J

Group No: 06

COACHING INSTITUTE MANAGEMENT SYSTEM

A software Engineering project submitted
By

Mazumder Urmila(18-36549-1)(Urmila-Mazumder)
Hossain,Muntasir (18-36421-1)(eraj01788)
Borno,Md.Sirajuddin (18-36449-1)(Born0)
Najbin,Sanjida (17-35722-3)(Sabiha722)

The project will be evaluated for the following Course Outcomes

CO3: Choose appropriate software engineering model in a software development environment	Total Marks
Project Background Analysis (needs, goal, benefits, etc.) [5Marks]	
Appropriate Process Model Selection [5Marks]	
Argumentation for model selection with Evidence [5Marks]	
Completeness, Spelling, Grammar and Organization of the Answer [5Marks]	
CO4: Explain the roles and their responsibilities in the software project management activities	Total Marks
Content Knowledge (e.g. System Requirements, System Design) [5Marks]	
Project Role identification [5Marks]	
Responsibility Description [5Marks]	
Completeness, Spelling, grammar and Organization of the Answer [5Marks]	

Contents

1. PROBLEM DOMAIN

1.1 Background to the Problem

This software helps manage Student, Teacher, Courses, Batches, Fees, Attendance and Marks. Provides easy reports like Outstanding Fees, Marks, Attendance etc. This is a must have software for all coaching institutes.

In the age of computing, we manage our coaching class through physical register or in excel sheets which is very difficult. When students or parents try to know about their results or any other information it’s really hard to find from register or excel sheet.

1.2 Solution to the Problem

The solutions I am going to propose to deal with this problem is coaching institute management system. Which can solve all these problems. It can receive data at a time and can provide the data also. In a coaching institute we saw that there are different people for different work but only a management system can solve all these problems. This solution is feasible to the meet the business objectives.

Our software name is Coaching Institute Management System which is developed after collecting requirements from several institutes in different fields. The software guarantees saving money and time for owners apart from organizing the institute information. There are many existing software solutions are available like school management system, coaching care but they are no so efficient as ours.

1.3 Existing / Related Solutions

There are many existing software solutions are available like school management system, coaching care. They only provide their system in their country or area. For this reason, another area can’t use this software.

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model

Software Development Cycle Process Model Analysis of nature and environment of the of the Software. It is a application made student management. The system parts have several kinds of interactions and those interactions are both serial and concurrent. The system needs to communicate with the database to functioning. It is a complex but less dynamic application based on database where many things are being done at the same time. Besides the system can be divided into several mini systems. So, we need to choose those software development process that are good for stable environment, allows incremental release of the system. We also can choose agile process that is lightweight, requires small group.

So by analyzing the nature and environment of our software and taking our group size and experience in consideration, we can say the best suited methods to develop our software would be incremental development of plan driven process and extreme programming of agile development process.

Incremental process model is also known as Successive version model. First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.

We already mentioned that our system can be divided into several mini systems. Among them the admitting student module gets the most priority of the students as it focuses on the problem domain most and benefits them as this will draw more students to them. Besides, if we can deliver the admission module first then we will be able to establish a core base for the whole system fast and gain the customers trust. After developing this module changing Info and then result publish module. So, clearly, we need to follow development process that allows to deliver project at increments. Then if we come to customers' requirements, they are well understood to us and there is low chance of change of requirements from client side. Furthermore, the environment of our selected software is stable.

Software Engineering Project: Student Management System

So, from these analyses we can say incremental model would be one of the best choices for small and inexperienced group like us. Choosing this model our risk of failing the overall project will reduce.

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development. Though our group is small and inexperienced, there is option of having high experienced personnel with us who can help us in all aspects. So, that way we can also use agile method to increase our productivity, output of incremental model, and we also can gather more experience. From the agile methods XP is the best suited option for us as it requires small group, allows incremental release, prioritization, simple design. It will also maximize our development and testing speed through pair programming. Choosing XP over other agile process is best for us because other process requires more manpower, tools, high experience, frequent professional meetings which are out of our reach.

2.2 Project Roll Identification and Responsibilities

Developing a software from scratch is a lengthy process and requires a lot of effort of multiple people. To finish a project successfully by avoiding project failure and produce a quality maintained software we need to follow a developing process that suits us most. Keeping that in mind we have already choose two process to develop our software. Now to carry out those process we need to divide all the task of the software's development life cycle and assign them to the members of the team. Who will perform which set of tasks are determined by the role of that person in project management and development. Those roles and their responsibilities (in our chosen process) can be:

- Customer: Customer is the source of funding. Pleasing him is all that matters. He writes the requirements and functional tests. He also can prioritizes the requirements. He is the one who decides if a functional requirement is satisfactory or out of the goal

- Programmer: They are responsible for writing the source code of the project. Our core team will mainly work as programmer or developer as we lack experience.

- Tester: Testers test the software for errors, bugs. They also help the customer to write functional test. They are driven by quality rather than delivery time. We can use our associates to test our system as they will be less bias than us.

- Tracker: Tracker traces the estimates made by the team and gives feedback on how accurate they are in order to improve future estimations. He also traces the progress of each iteration and evaluates whether the goal is reachable within the given resource and time constraints or if any changes are needed in the process. In our project, he must me one of our experienced associates.

- Consultant: Consultant is not directly involved with the development or management. Rather, he is an external member who can provide specific technical knowledge if any directly involved member need any veterans suggestion. If anyone of us needs any consultation of a domain we may consult a highly experienced personnel of that particular domain.

- Manager: Manager is the big boss. He makes the big decisions of the project. Again we may take help from a highly experienced personnel as making critical decisions require high estimation and past experience.

3. PRODUCT AND PROJECT DESCRIPTION

3.1 Stakeholders

Admin: Admin can maintain the entire system.

Manager: Manager manages students, Batch, Course.

UML designing member.

Project implementation members.

Database handling members

Internal and external stakeholders:

Internal: Admin, Manager, Developer

External: Consultant, User

3.2 System Features/ Functional Requirements

1. Login
2. Add new student
3. View student list
4. View batch
5. View course list
6. View payment info
7. View student result
8. Add new batch
9. Change batch time name
10. Student edit his/her info
11. Upload student’s marks
12. Add new course
13. Send mail or message.

4. System Quality Attributes

Usability: A trained user shall be able to admit a new student within 2 to 4 min

4.1 System Architecture

Include use case diagram, class diagram, activity diagram, state-chart diagram, ER diagram, Data Flow diagram, etc.

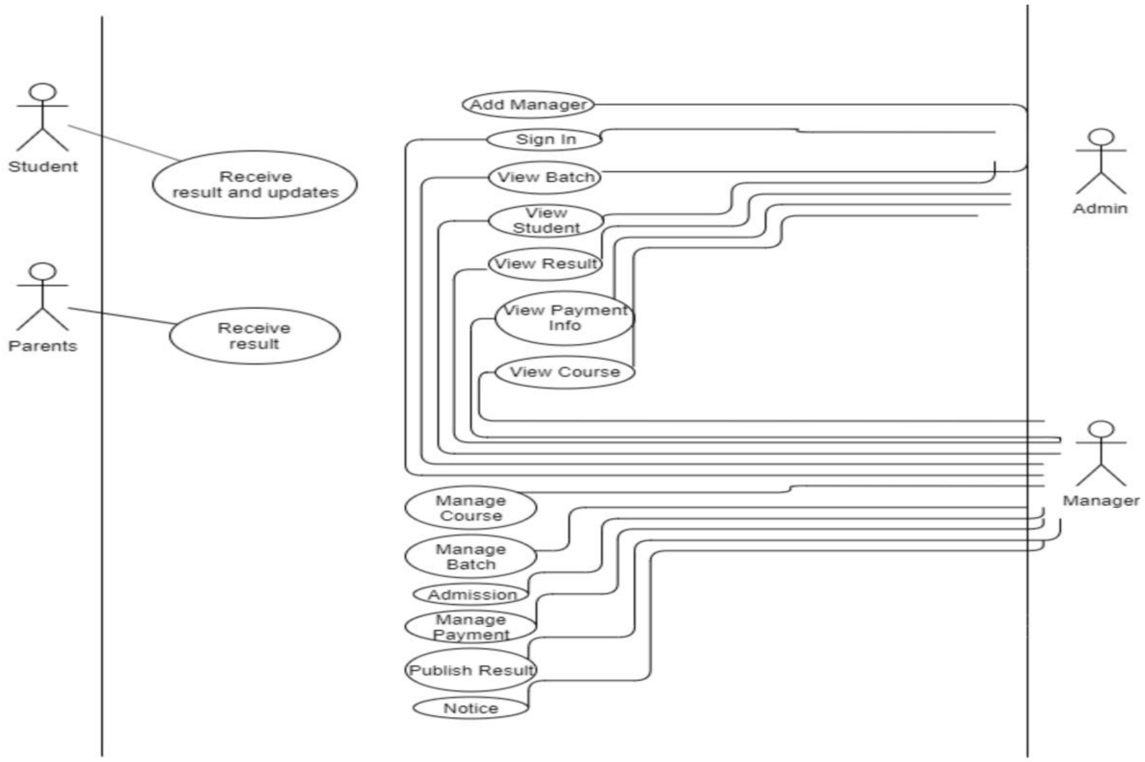


Figure: Use Case Diagram

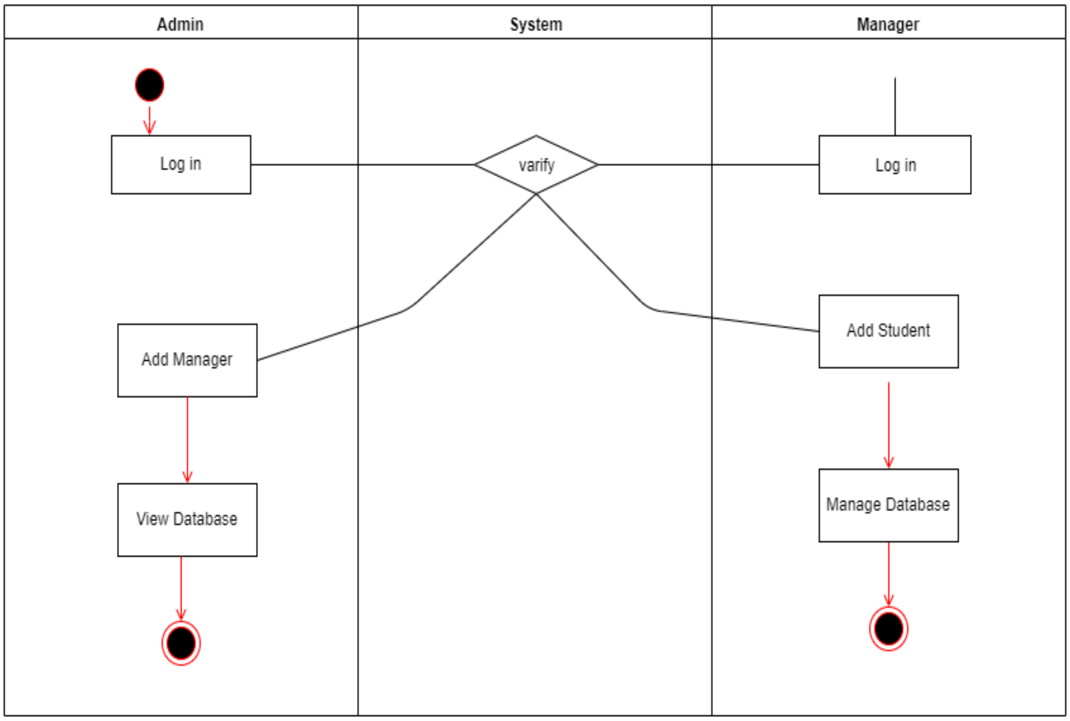


Figure: Activity Diagram

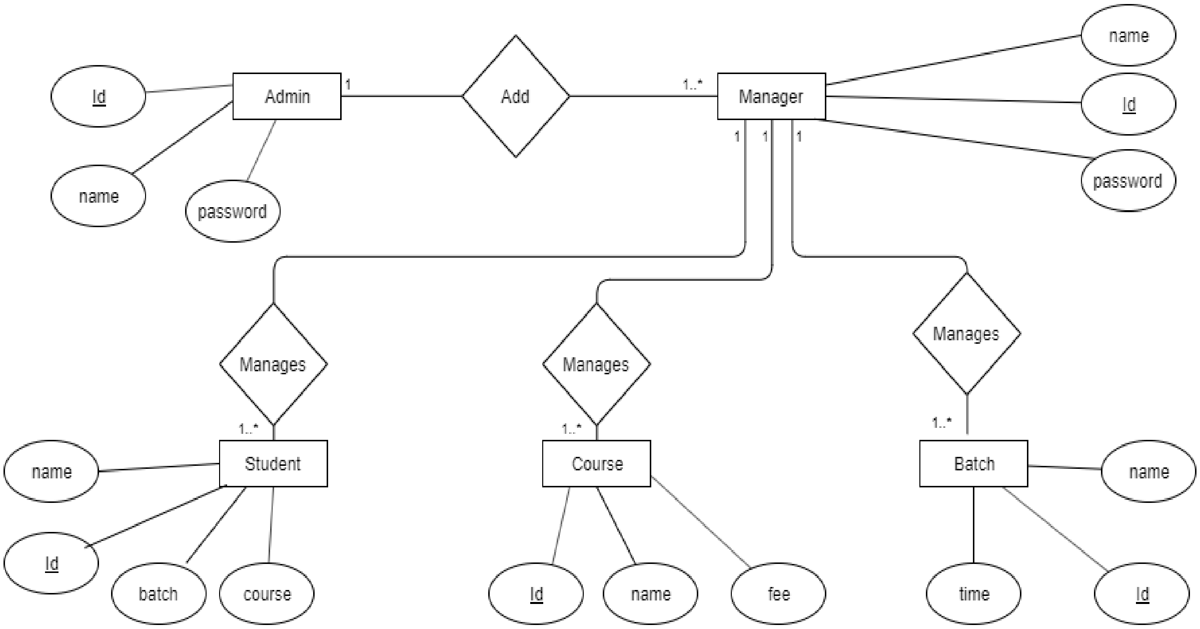


Figure: E-R Diagram

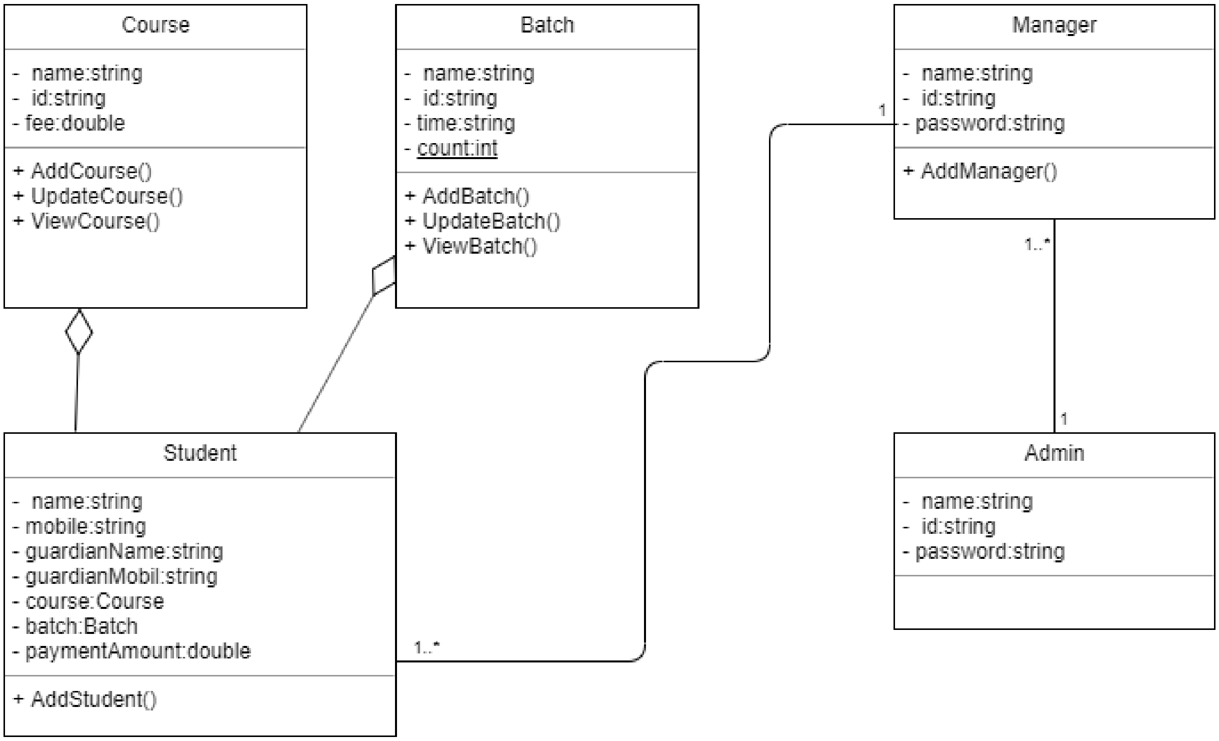


Figure: Class Diagram

Class Name

- The name of the class appears in the first partition.

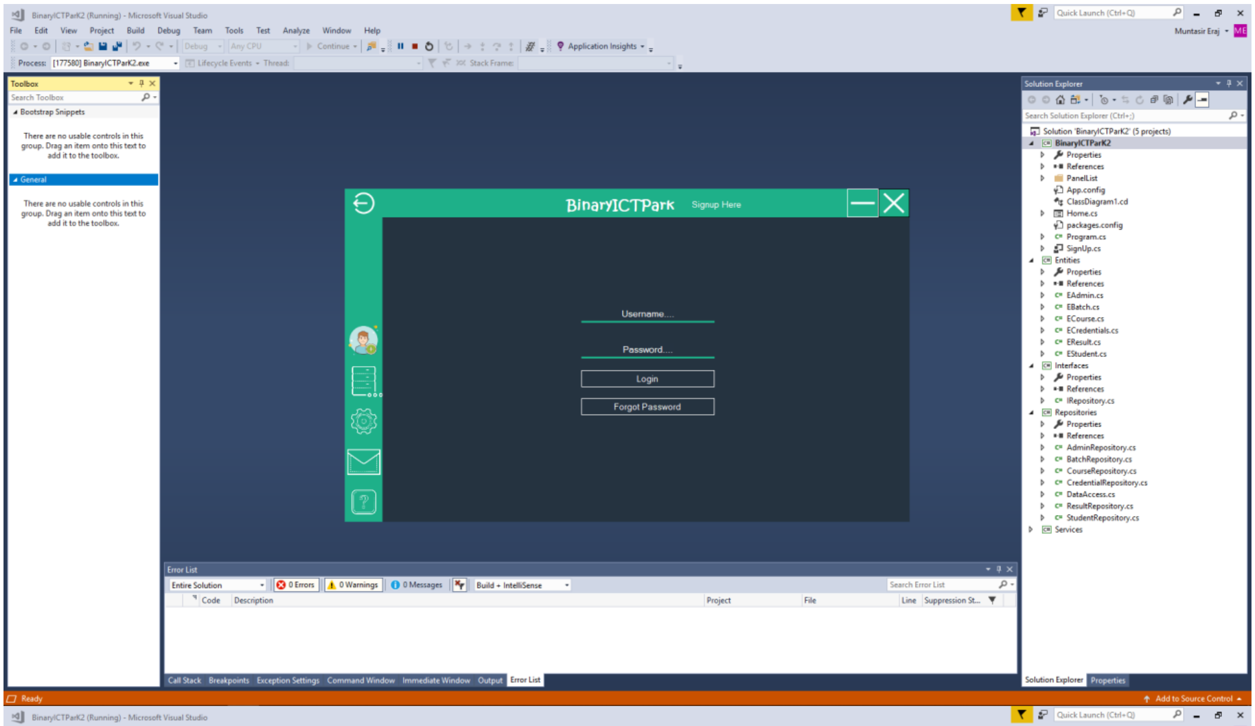
Class Attributes

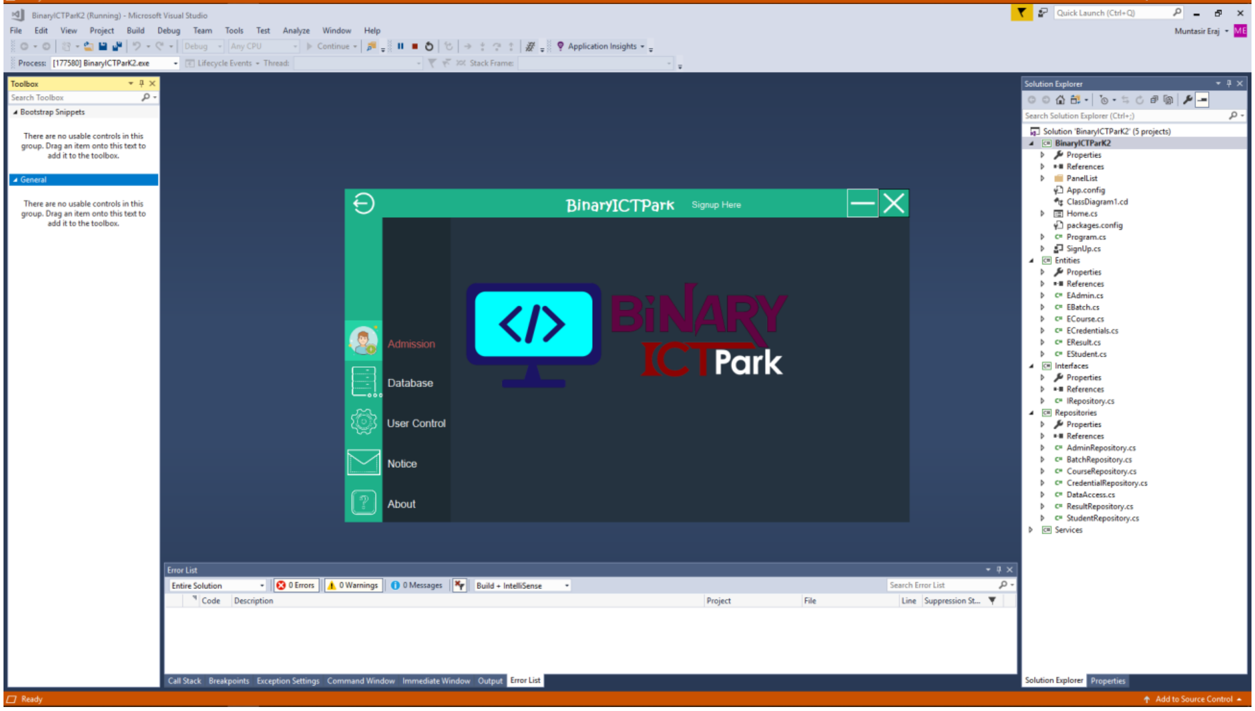
- Attributes are shown in the second partition.
- The attribute type is shown after the colon.
- Attributes map onto member variables (data members) in code.

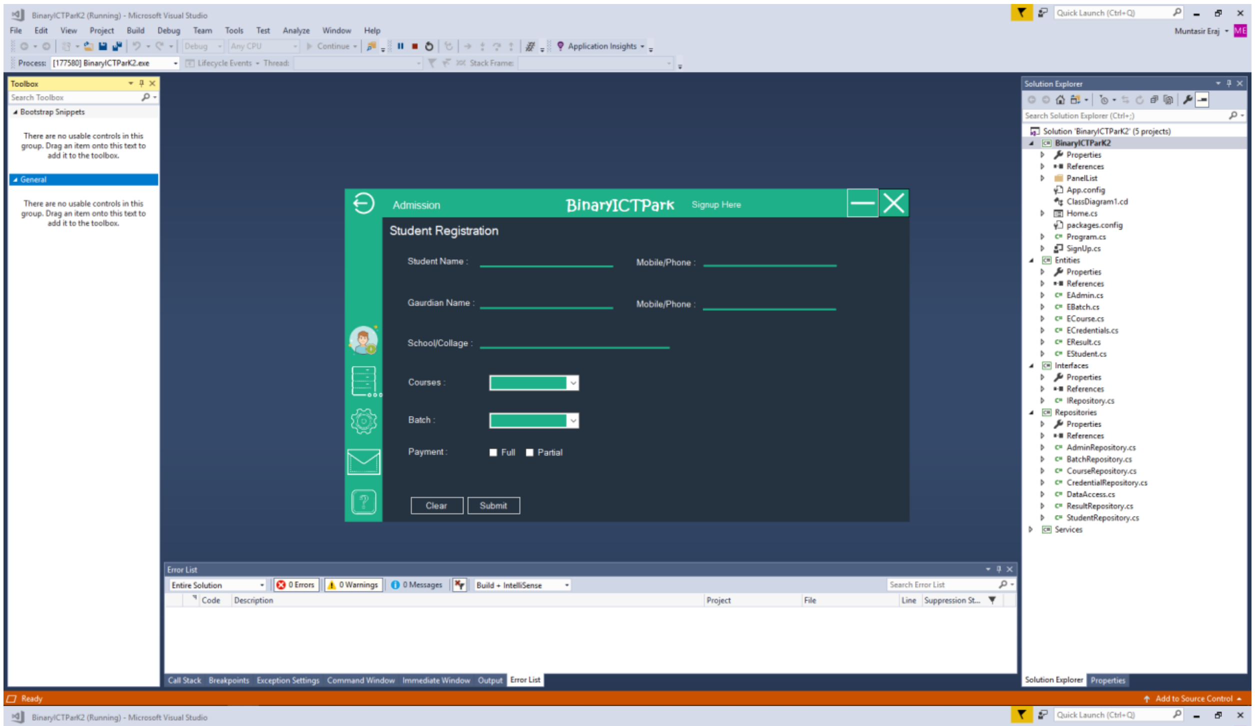
Class Operations (Methods)

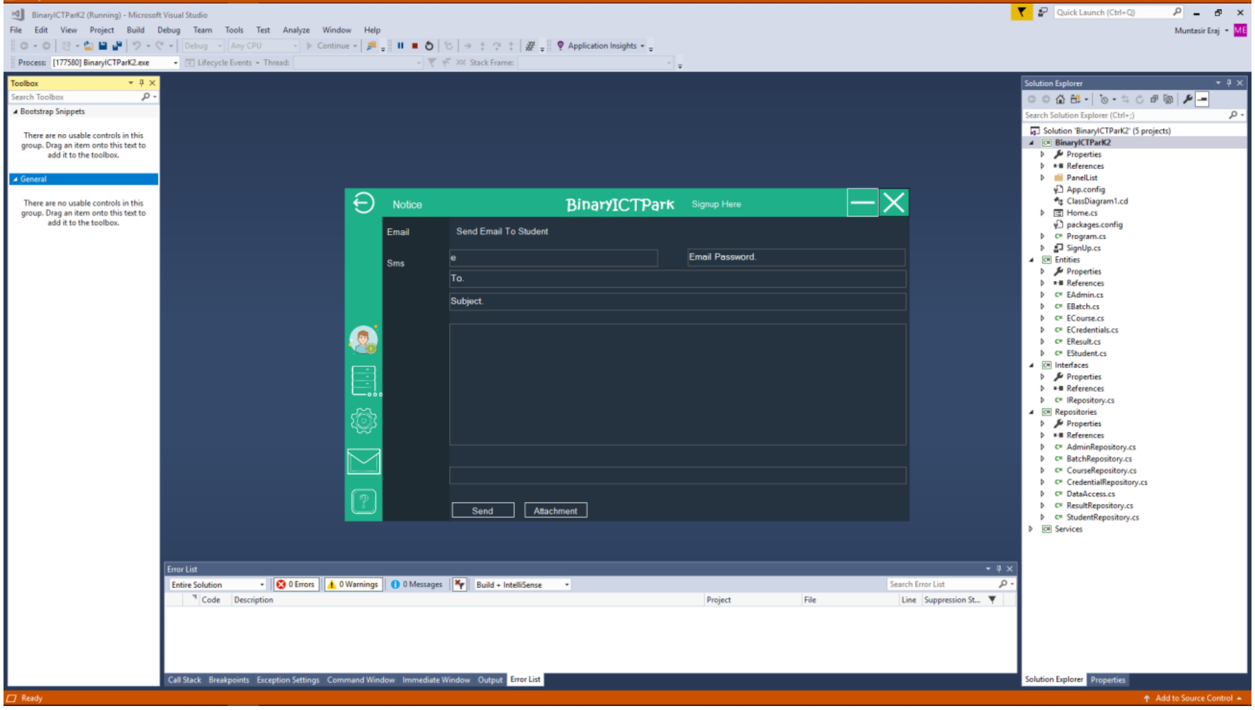
- Operations are shown in the third partition. They are services the class provides.
- The return type of a method is shown after the colon at the end of the method signature.
- The return type of method parameters is shown after the colon following the parameter name.
- Operations map onto class methods in code

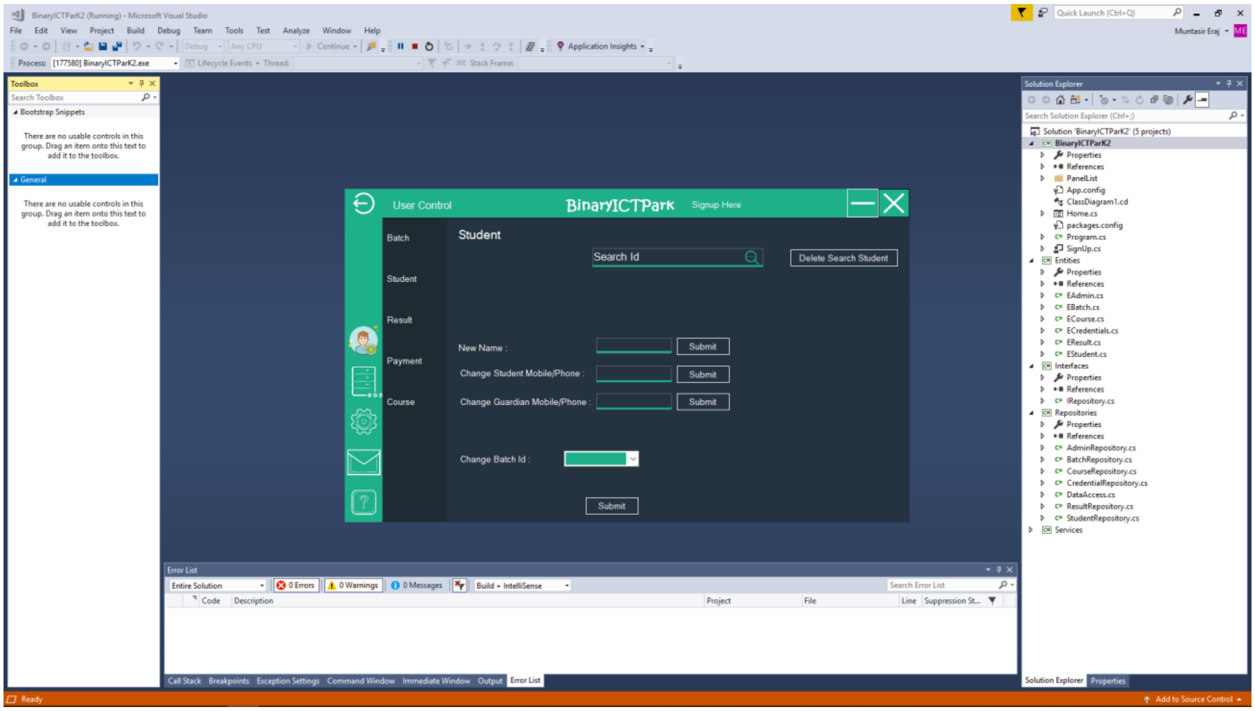
4.2 System Interface






BinaryICTPark2 (Running) - Microsoft Visual Studio
Toolbox
Solution Explorer
Admission
BinaryICTPark
Signup Here
Student Registration
Student Name :
Mobile/Phone :
Gaurdian Name :
Mobile/Phone :
School/Collage :
Courses :
Batch :
Payment :
□ Full
□ Partial
Clear
Submit
Error List
0 Errors
0 Warnings
0 Messages
Ready


BinaryICTPark2 (Running) - Microsoft Visual Studio
Toolbox
Solution Explorer
Notice
BinaryICTPark
Signup Here
Email
Sms
Send Email To Student
e
Email Password.
To.
Subject.
Send
Attachment
Error List
0 Errors
0 Warnings
0 Messages
Ready



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

4.3 Project Requirements

Loc=8600

Effort: $E=a_b\,(KLoc)^{bb}$

$=2.4(8.6K)^{1.05}$

$=22.98$ PM

Time: $D=c_b(E)^{db}$

$=2.5(22.98)^{.38}$

$=8$ PM

Resource: $P=E/D$

$=22.98/8 \quad = 2.87\ (3)$

Budget (Static Single Variable Model)

Cost $C=a(L)^{b}$

$E=1.4(8600)^{.93}$

$=6385.74$ PM

$Doc=30.4(8600)^{.9} \quad =105662.81$

$D=4.6(8600)^{.26} \quad =48.4$

SCHEDULING:

	Week-1	Week-2	Week-3	Week-4	Week-5	Week-6	Week-7	Week-8	Week-9	Week-10	Week-11	Week-12
Meeting	■											
Design		■	■									
Coding				■	■	■	■	■	■			
Testing					■	■	■	■	■	■		
Release											■	
Documentation												■

REFFERENCE:

[1]GreeksforGreeks. (2017,Sep.) geeksforgeeks.org[Online]. Available: https://www.geeksforgeeks.org/software-engineering-incremental-process-model/. [Accessed Dec.14, 2019].

[2]AgileAlliance. (2016,Aug.) agilealliance.org[Online]. Available: https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1). [Accessed Dec.14, 2019].

[4]R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*

[5]Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.

[6]Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994).In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.