

Academic Analytics System

- SEMESTER: SPRING 2024
- COURSE: CS 590 (FUNDAMENTALS OF SE)
- PROFESSOR: DR. VLADIMIR USKOV
- TEAM: CS590_TEAM_01
- TEAM MEMBERS: LIKHITHA KRISHNA CHERUKURI
PAVAN KALYAN GARIKAPATI
NITHIN BHARATH REDDY VASIPALLI



Table of Contents

S.No	Content	Slide No.
1	Introduction and Project Goal	5
2	Objectives	6
3	Project Outcomes	7
4	Perspective SE Model	8
5	10 Academic Analytics Systems for Top Universities	9
6	15 Common Data/Class Objects	10
7	5 Proposed New Data/Class Objects	11
8	Summary Table for Class/Data Objects	12 – 13
9	15 Common Functions	14
10	5 Proposed New Functions	15
11	Summary Table for Functions	16 – 17
12	Software Engineering Diagrams	18

S.No	Content	Slide No.
13	Use Case Diagram Using UML GUI – Screen Capture	19
14	Use Case Diagram Using UML - PDF	20
15	Class Object Diagram Using IBM Rational GUI – Screen Capture	21 – 22
16	Class Object Diagram Using IBM Rational - PDF	23 – 24
17	Entity Relationship Diagram Using Smart Draw – Screen Capture	25
18	Context Data Flow Diagram Using Smart Draw GUI – Screen Capture	26
19	Context Data Flow Diagram Using Smart Draw – PDF	27
20	Legends For Context Data Flow Diagram	28 – 30
21	Level-0 Data Flow Diagram Using VISIO GUI – Screen Capture	31
22	Level-0 Data Flow Diagram Using VISIO – PDF	32
23	Legends For Level-0 Data Flow Diagram	33 – 34

S.No	Content	Slide No.
24	State Transition Diagram Using Visual Paradigm GUI – Screen capture	35
25	State Transition Diagram Using Visual Paradigm - PDF	36
26	Software Architectural Model	37 - 39
27	Software System Hierarchical Design Model Using Visio GUI - Screen Capture	40 – 42
28	Software System Hierarchical Design Model Using Visio - PDF	43 - 45
29	UE's Hierarchical Design – Functional Model GUI - Screen Capture	46
30	UE's Hierarchical Design – Functional Model - PDF	47
31	UE's Hierarchical Design - Components and Design Model GUI - Screen Capture	48
32	UE's Hierarchical Design - Components and Design Model - PDF	49
33	LOC Based Project Cost Estimation	50
34	FP Based Project Cost Estimation	51
35	Conclusion	52

Introduction

Academic Analytics Systems are essential tools for modern universities, facilitating data-driven decision-making and enhancing educational outcomes. These systems integrate and analyze various academic data, from student performance to institutional procedures, using predictive analytics to support student success and optimize resource allocation. By tracking student progress, identifying at-risk students, and improving communication between faculty and students, these systems empower educators to personalize interventions and refine curriculum strategies. Additionally, they prioritize data security measures to protect sensitive information and ensure compliance with privacy regulations, fostering trust among stakeholders. Globally adopted, Academic Analytics Systems play a vital role in enhancing decision-making processes and driving educational excellence.

In summary, Academic Analytics Systems streamline academic processes, support student success, and empower institutions to make informed decisions. By leveraging predictive analytics, these systems enable universities to proactively address challenges, refine educational strategies, and foster a culture of data-driven innovation, ultimately enhancing the overall educational experience for students and faculty alike.

Project Goal

The Academic Analytics System project seeks to develop a tailored software solution for software engineering education. By integrating data analytics, predictive modeling, and resource optimization, the system aims to enhance student learning outcomes, support faculty in personalized interventions, and optimize institutional processes. Ultimately, the goal is to empower educational institutions to make data-driven decisions, foster student success, and elevate the quality of software engineering education.

Objectives

1. **Data Integration:** Develop mechanisms to seamlessly gather and integrate diverse data sources relevant to software engineering education, including student enrollment, academic performance, and faculty information.
2. **Predictive Analytics:** Implement predictive modeling algorithms to forecast academic trends, identify at-risk students, and provide timely interventions to support student success and retention.
3. **Real-time Monitoring:** Enable real-time monitoring of student progress, including attendance, assignment submissions, and exam scores, to identify areas for improvement and facilitate personalized interventions.
4. **Enhanced Communication:** Improve communication between faculty and students by providing accessible performance data, enabling students to take an active role in their academic journey, and fostering a collaborative learning environment.
5. **Customizable Dashboards:** Develop customizable dashboards to provide educators and administrators with a comprehensive overview of institutional metrics, facilitating informed decision-making and strategic planning.
6. **Curriculum Refinement:** Analyze academic data to refine curriculum strategies, ensuring alignment with emerging educational trends and industry demands, thereby enhancing the overall learning experience for students.

Project Outcomes

- 1. Improved Student Success:** By leveraging predictive analytics, the project aims to identify at-risk students early and provide timely interventions, ultimately leading to improved student retention and success rates.
- 2. Optimized Resource Allocation:** Through data-driven analysis of factors such as class sizes and faculty workload, the project seeks to optimize resource allocation, ensuring efficient use of resources and enhancing overall institutional effectiveness.
- 3. Enhanced Teaching Effectiveness:** Real-time monitoring of student progress and personalized interventions enable faculty members to tailor their teaching approaches to meet individual student needs, fostering a more effective learning environment.
- 4. Streamlined Decision-Making:** The implementation of customizable dashboards and data analytics tools empowers administrators and educators to make informed decisions, leading to more effective strategic planning and resource management.
- 5. Data Security and Compliance:** Implementation of robust data security measures ensures the protection of sensitive academic information, fostering trust among stakeholders and maintaining compliance with privacy regulations.

Perspective SE Process Model

For the development of Academic Analytics System, Spiral Software Development Life Cycle(SDLC) model is used.

The Spiral Model is indeed well-suited for the development of the Academic Analytics System due to its unique characteristics and the specific requirements of such a project. The Spiral Model emphasizes risk management by incorporating iterative development cycles with risk analysis and mitigation activities at each phase.

Here's how the Spiral Model aligns with the needs and complexities of developing an Academic Analytics System:

- 1.Risk Management
- 2.Iterative Prototype
- 3.Flexibility
- 4.Phased Approach



10 Academic Analytics Systems for Top Universities

1. MIT - Massachusetts Institute of Technology
2. Columbia University
3. Stanford University
4. University of Chicago
5. Yale University
6. University of California, Berkeley
7. Georgia Institute of Technology
8. Purdue University
9. Texas A&M University
10. Harvard university

15 Common Data/Class Objects

- Financial Aid Manager
- Student Information Systems
- Data Warehouse
- Mobile Application
- Learning Management System
- Internship Programs
- Alumni Coordinator
- Data Visualisation System
- Social Media Network
- Curriculum Schema
- Student Survey System
- Access Logs
- Course Catalog
- Library Repository
- Course Feedback System

5 Proposed New Data/Class Objects

Faculty Development Programs

Student Success Tracker

Ethics and Compliance Auditor

Academic Remainder System

Retention Rate System

Summary Table for Class/Data Objects

University Websites	Identified Class Objects/ Data Objects	Massachusetts Institute of Technology	Columbia University	Stanford University	University of Chicago	Yale University	University of California, Berkeley	Georgia Institute of Technology	Purdue University	Texas A&M University	Harvard University	No. of Universities Used
Data/Class Objects Identified in Harvard University	Financial Aid Manager	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
	Alumni Coordinator	+	+	+		+	+	+			+	7 out of 10 (70%)
Data/Class Objects Identified in Massachusetts Institute of Technology	Data Warehouse	+	+	+				+		+	+	6 out of 10 (60%)
	Course Feedback System	+			+		+		+	+	+	6 out of 10 (60%)
	Curriculum Schema	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
Data/Class Objects in Stanford University	Social Media Network	+	+	+			+	+		+	+	7 out of 10 (70%)
	Student Information Systems	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
Data/Class Objects in University of Chicago	Data Visualization System	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
	Access Logs	+	+	+	+		+	+			+	7 out of 10 (70%)

Summary Table for Class/Data Objects

15 Common Functions

- Manage Financial Aid Disbursement
- Co-ordinate Student Records
- Monitor Academic Integrity
- Develop features for Mobile Application
- Manage Database
- Evaluate Student Learning Outcomes
- Notify Internship Opportunities
- Maintain Alumni Records
- Develop New Connections
- Plan Curriculum
- Visualize Data
- Review Feedback Data
- Monitor Access Information
- Manage Course Information
- Manage Library Acquisition

5 Proposed New Functions

Evaluate Faculty Performance

Track Student Performance

Review Ethical Policies

Implement Early Alert System

Analyze Retention Trends

Summary Table for Functions

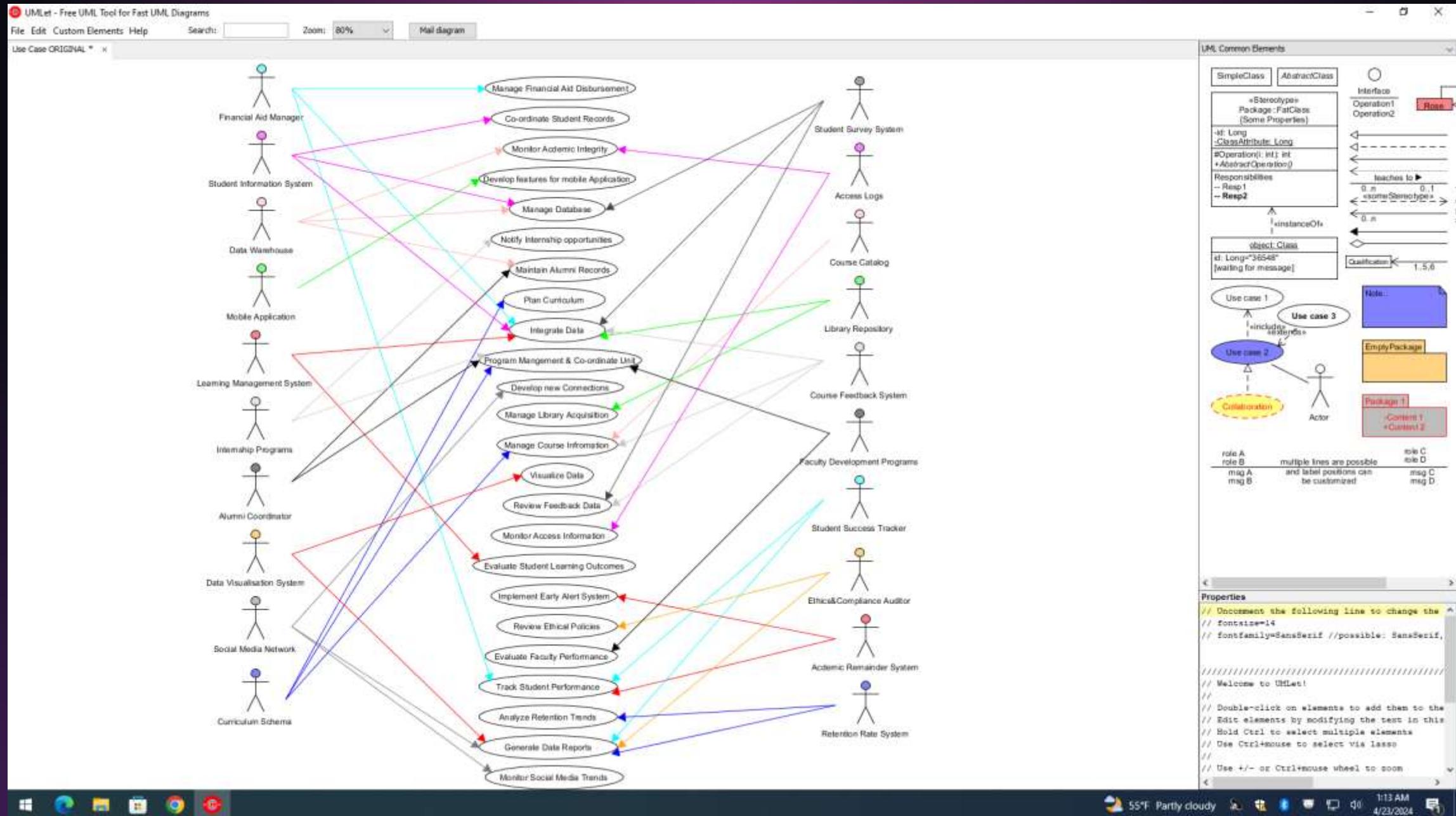
University Websites	Identified Functions	Massachusetts Institute of Technology	Columbia University	Stanford University	University of Chicago	Yale University	University of California, Berkeley	Georgia Institute of Technology	Purdue University	Texas A&M University	Harvard University	No. of Universities Used
Functions Identified in Harvard University	Manage Financial Aid Disbursement	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
	Maintain Alumni Records	+	+	+		+	+	+			+	7 out of 10 (70%)
Functions Identified in MIT	Manage Database	+	+	+				+		+	+	6 out of 10 (60%)
	Manage Library Acquisition	+			+		+		+	+	+	6 out of 10 (60%)
	Plan Curriculum	+	+	+	+	+	+	+	+	+	+	10 out of 10 (100%)
Functions Identified in Stanford University	Monitor Access Information	+	+	+			+			+	+	7 out of 10 (70%)
	Co-ordinate Students Records Management	+	+	+	+	+	+	+	+	+	+	10 out 10 (100%)
Functions Identified in University of Chicago	Visualize Data	+	+	+	+	+	+	+	+	+	+	10 out of 10(100%)
	Monitor Academic Integrity	+	+	+	+		+	+			+	7 out 10 (70%)

Summary Table for Functions

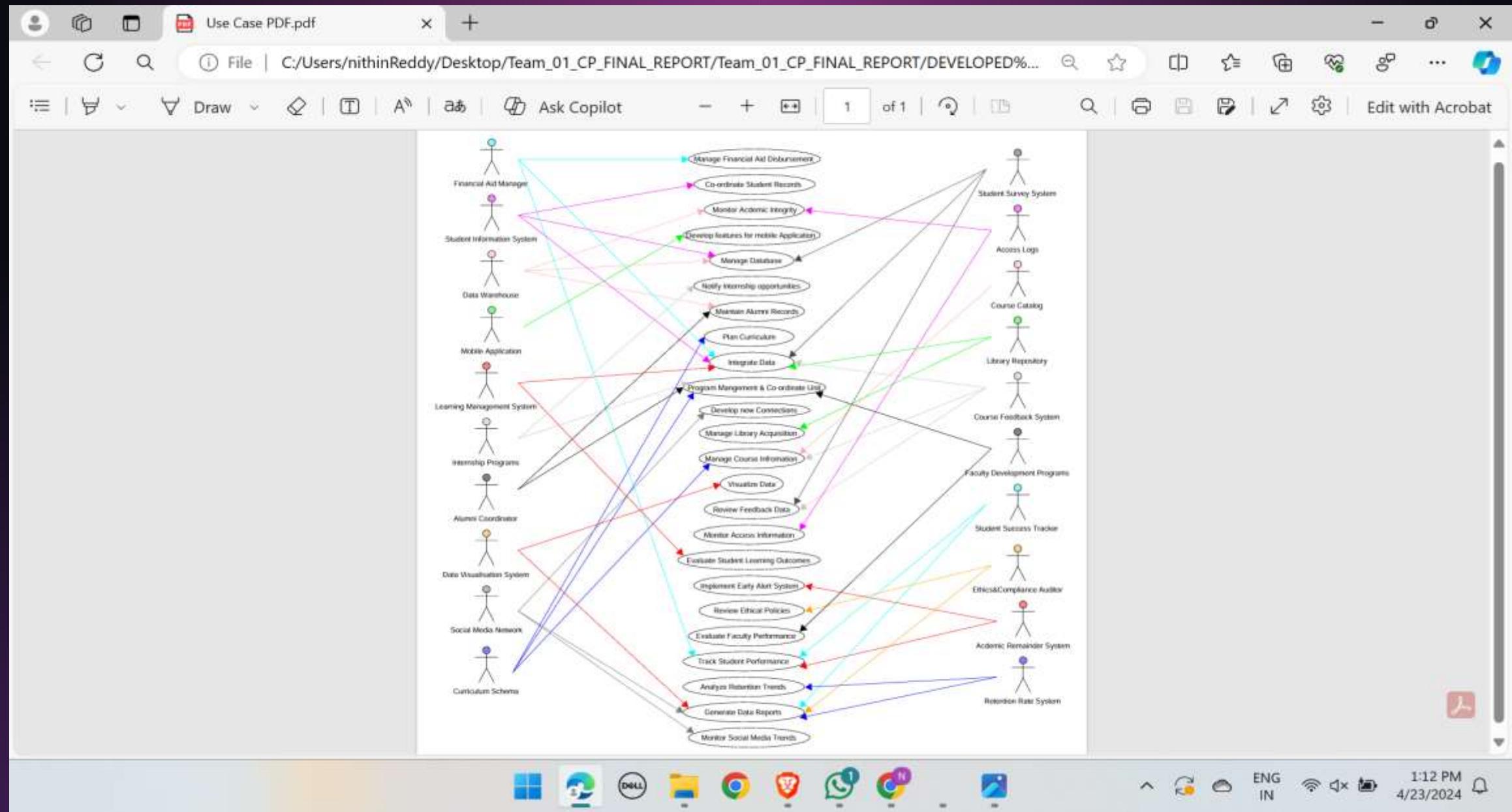
SOFTWARE ENGINEERING DIAGRAMS

- Use Case Diagram
- Class Object Diagram
- Entity Relationship Diagram
- Context Data Flow Diagram
- Level – 0 Data Flow Diagram
- State Transition Diagram
- Software Architectural Model
- Software System Hierarchical Design Model
- User Experience Functional Model
- User Experience Components and Design Model

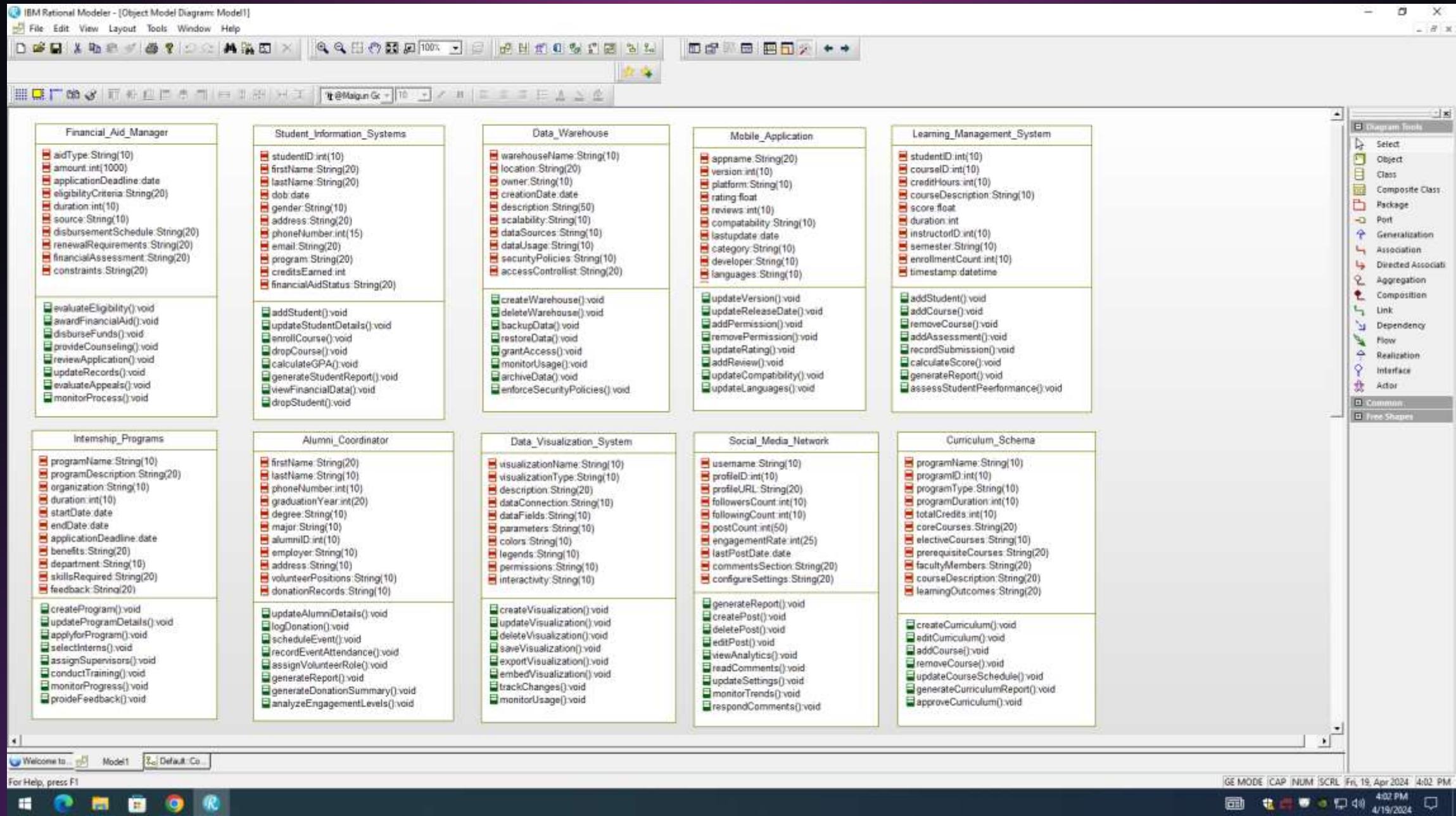
Use Case Diagram Using UM Let GUI – Screen Capture



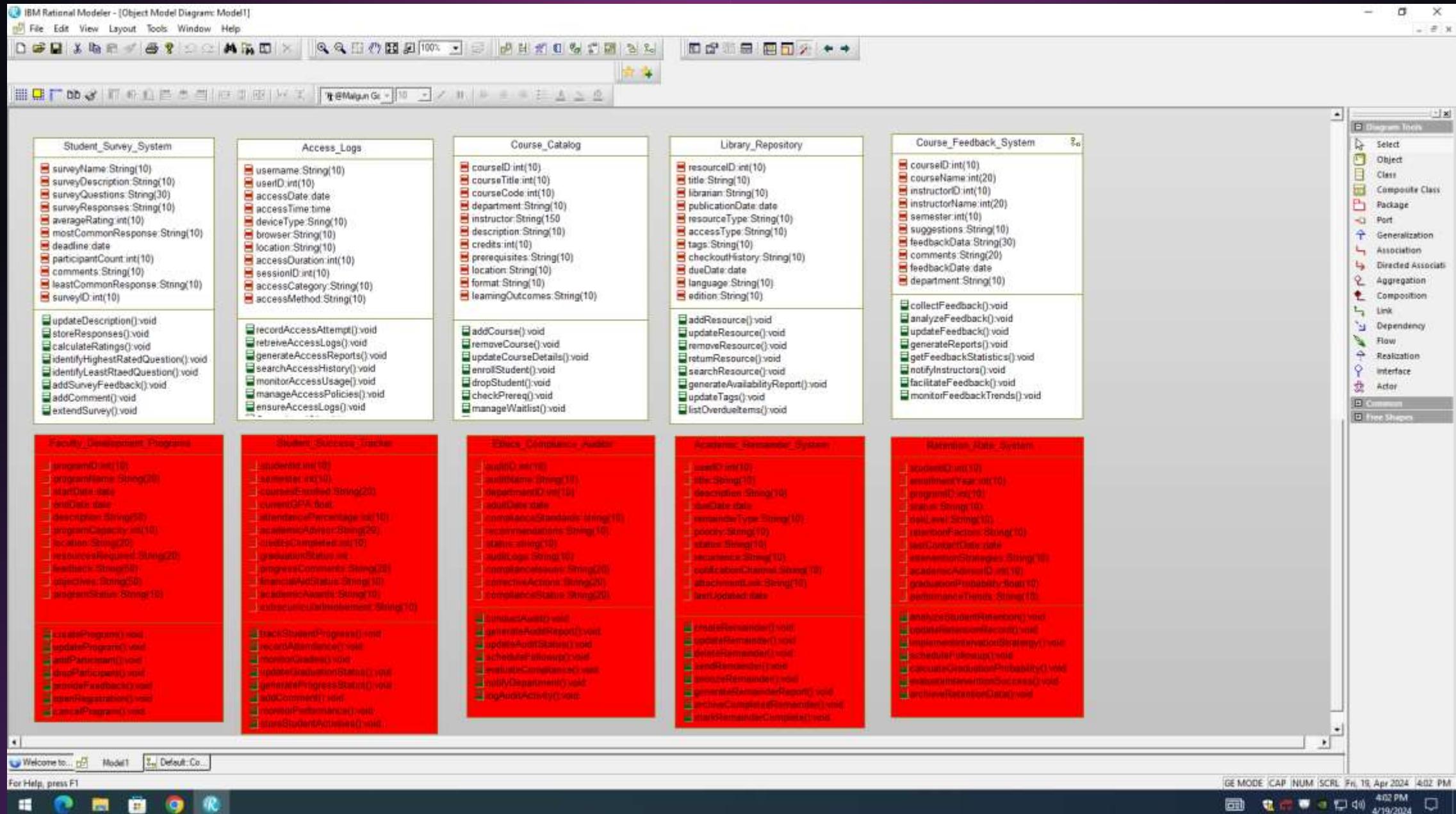
Use Case Diagram Using UM Let - PDF



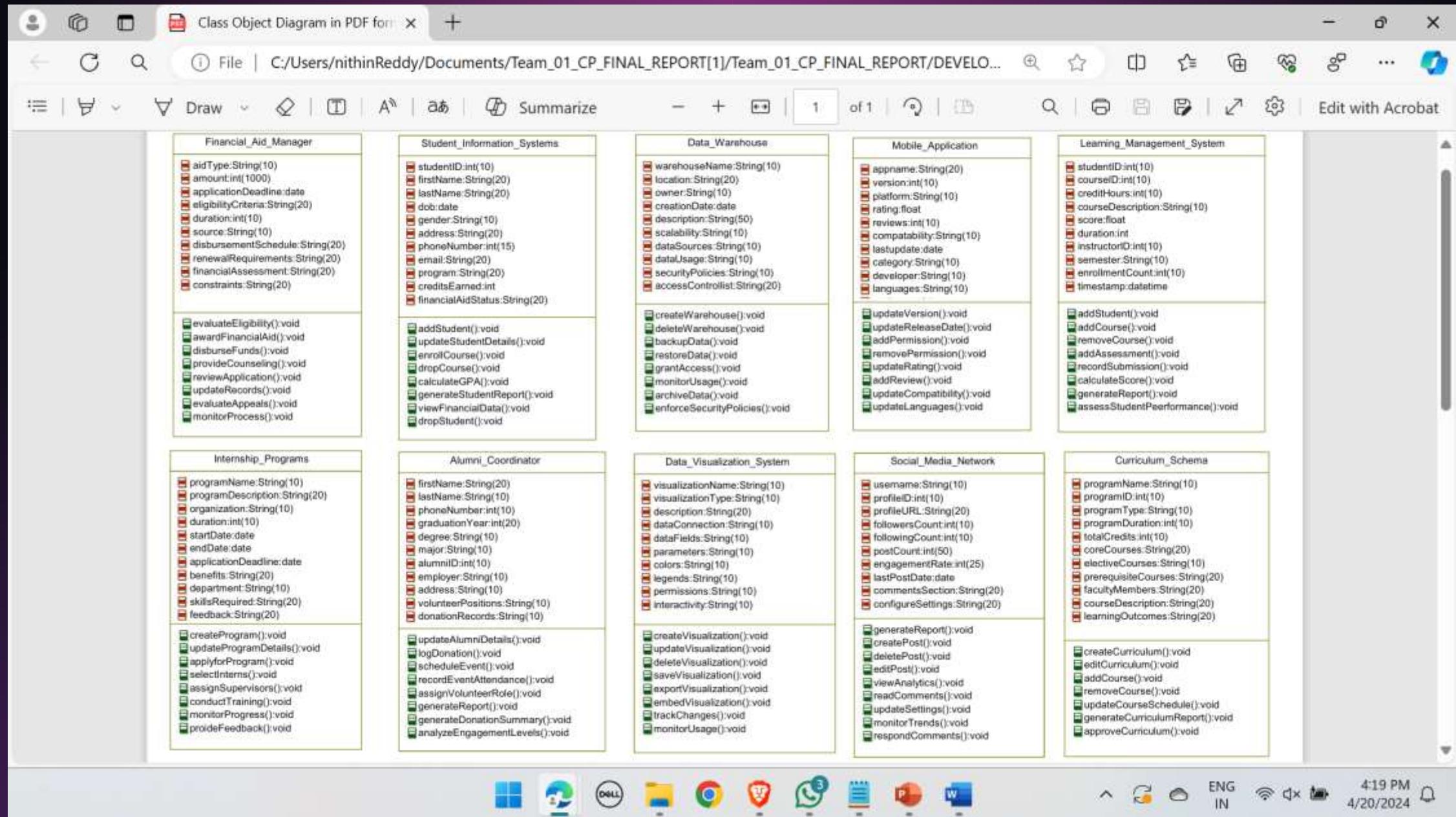
Class Object Diagram Using IBM Rational GUI – Screen Capture



Class Object Diagram Using IBM Rational GUI – Screen Capture



Class Object Diagram Using IBM Rational - PDF



Class Object Diagram Using IBM Rational - PDF



Class Object Diagram in PDF format

File | C:/Users/nithinReddy/Documents/Team_01_CP_FINAL_REPORT[1]/Team_01_CP_FINAL_REPORT/DEVELOPMENT/Class_O

Draw | Summarize | 1 of 1 | Edit with Acrobat

Student_Survey_System

- Attributes:
 - surveyName: String(10)
 - surveyDescription: String(10)
 - surveyQuestions: String(30)
 - surveyResponses: String(10)
 - averageRating: int(10)
 - mostCommonResponse: String(10)
 - deadline: date
 - participantCount: int(10)
 - comments: String(10)
 - leastCommonResponse: String(10)
 - surveyID: int(10)
- Operations:
 - updateDescription(): void
 - storeResponses(): void
 - calculateRatings(): void
 - identifyHighestRatedQuestion(): void
 - identifyLeastRatedQuestion(): void
 - addSurveyFeedback(): void
 - addComment(): void
 - extendSurvey(): void

Access_Logs

- Attributes:
 - username: String(10)
 - userID: int(10)
 - accessDate: date
 - accessTime: time
 - deviceType: String(10)
 - browser: String(10)
 - location: String(10)
 - accessDuration: int(10)
 - sessionID: int(10)
 - accessCategory: String(10)
 - accessMethod: String(10)
- Operations:
 - recordAccessAttempt(): void
 - retrieveAccessLogs(): void
 - generateAccessReports(): void
 - searchAccessHistory(): void
 - monitorAccessUsage(): void
 - manageAccessPolicies(): void
 - ensureAccessLogs(): void

Course_Catalog

- Attributes:
 - courseID: int(10)
 - courseTitle: int(10)
 - courseCode: int(10)
 - department: String(10)
 - instructor: String(150)
 - description: String(10)
 - credits: int(10)
 - prerequisites: String(10)
 - location: String(10)
 - format: String(10)
 - learningOutcomes: String(10)
- Operations:
 - addCourse(): void
 - removeCourse(): void
 - updateCourseDetails(): void
 - enrollStudent(): void
 - dropStudent(): void
 - checkPrereq(): void
 - manageWaitlist(): void

Library_Repository

- Attributes:
 - resourceID: int(10)
 - title: String(10)
 - librarian: String(10)
 - publicationDate: date
 - resourceType: String(10)
 - accessType: String(10)
 - tags: String(10)
 - checkoutHistory: String(10)
 - location: String(10)
 - dueDate: date
 - language: String(10)
 - edition: String(10)
- Operations:
 - addResource(): void
 - updateResource(): void
 - removeResource(): void
 - returnResource(): void
 - searchResource(): void
 - generateAvailabilityReport(): void
 - updateTags(): void
 - listOverdueItems(): void

Course_Feedback_System

- Attributes:
 - courseID: int(10)
 - courseName: int(20)
 - instructorID: int(10)
 - instructorName: int(20)
 - semester: int(10)
 - suggestions: String(10)
 - feedbackData: String(30)
 - comments: String(20)
 - feedbackDate: date
 - department: String(10)
- Operations:
 - collectFeedback(): void
 - analyzeFeedback(): void
 - updateFeedback(): void
 - generateReports(): void
 - getFeedbackStatistics(): void
 - notifyInstructors(): void
 - facilitateFeedback(): void
 - monitorFeedbackTrends(): void

Faculty_Development_Programs

- Attributes:
 - programID: int(10)
 - programName: String(20)
 - startDate: date
 - endDate: date
 - description: String(50)
 - programCapacity: int(10)
 - education: String(20)
 - resourcesRequired: String(20)
 - feedback: String(50)
 - objectives: String(50)
 - programStatus: String(10)
- Operations:
 - createProgram(): void
 - updateProgram(): void
 - infoParticipant(): void
 - dropParticipant(): void
 - provideFeedback(): void
 - cancelRegistration(): void
 - cancelProgram(): void

Student_Success_Tracker

- Attributes:
 - studentID: int(10)
 - semester: int(10)
 - coursesEnrolled: String(20)
 - currentGPA: float
 - attendancePercentage: int(10)
 - academicAdvisor: String(30)
 - gradesCompleted: int(10)
 - graduationStatus: int
 - progressComments: String(20)
 - financialAidStatus: String(10)
 - academicAwards: String(10)
 - extraCurricularInvolvement: String(10)
- Operations:
 - trackStudentProgress(): void
 - recordAttendance(): void
 - monitorGrades(): void
 - updateGraduationStatus(): void
 - generateProgressStatus(): void
 - addComments(): void
 - monitorPerformance(): void
 - storeStudentActivities(): void

Ethics_Compliance_Auditor

- Attributes:
 - auditID: int(10)
 - auditorName: String(10)
 - department: String(10)
 - auditDate: date
 - complianceStandards: String(10)
 - recommendations: String(10)
 - status: String(10)
 - auditLog: String(10)
 - communications: String(20)
 - correctiveActions: String(20)
 - complianceStatus: String(20)
- Operations:
 - conductAudit(): void
 - generateAuditReport(): void
 - updateAuditStatus(): void
 - scheduleFollowup(): void
 - evaluateCompliance(): void
 - notifyDepartment(): void
 - logAuditActivity(): void

Academic_Reminder_System

- Attributes:
 - userID: int(10)
 - title: String(10)
 - description: String(10)
 - dueDate: date
 - reminderType: String(10)
 - priority: int(10)
 - status: String(10)
 - recurrence: String(10)
 - notificationChannel: String(10)
 - attachmentLink: String(10)
 - lastUpdated: date
- Operations:
 - createReminder(): void
 - updateReminder(): void
 - updateReminderRecord(): void
 - endRemainderInterval(): void
 - cancelReminder(): void
 - cancelRemainderReport(): void
 - archiveCompletedReminder(): void
 - markReminderComplete(): void

Retention_Rate_System

- Attributes:
 - studentID: int(10)
 - enrollmentID: int(20)
 - programID: int(10)
 - status: String(10)
 - retailLevel: String(10)
 - retentionFactors: String(10)
 - lastContactDate: date
 - interventionStrategies: String(10)
 - academicAdvisorID: int(10)
 - graduationProbability: float(10)
 - performanceTrends: String(10)
- Operations:
 - analyzeStudentPerformance(): void
 - updateRetentionRecord(): void
 - endRemainderInterval(): void
 - scheduleFollowup(): void
 - cancelGraduationProbability(): void
 - evaluateInterventionSuccess(): void
 - archiveRetentionData(): void

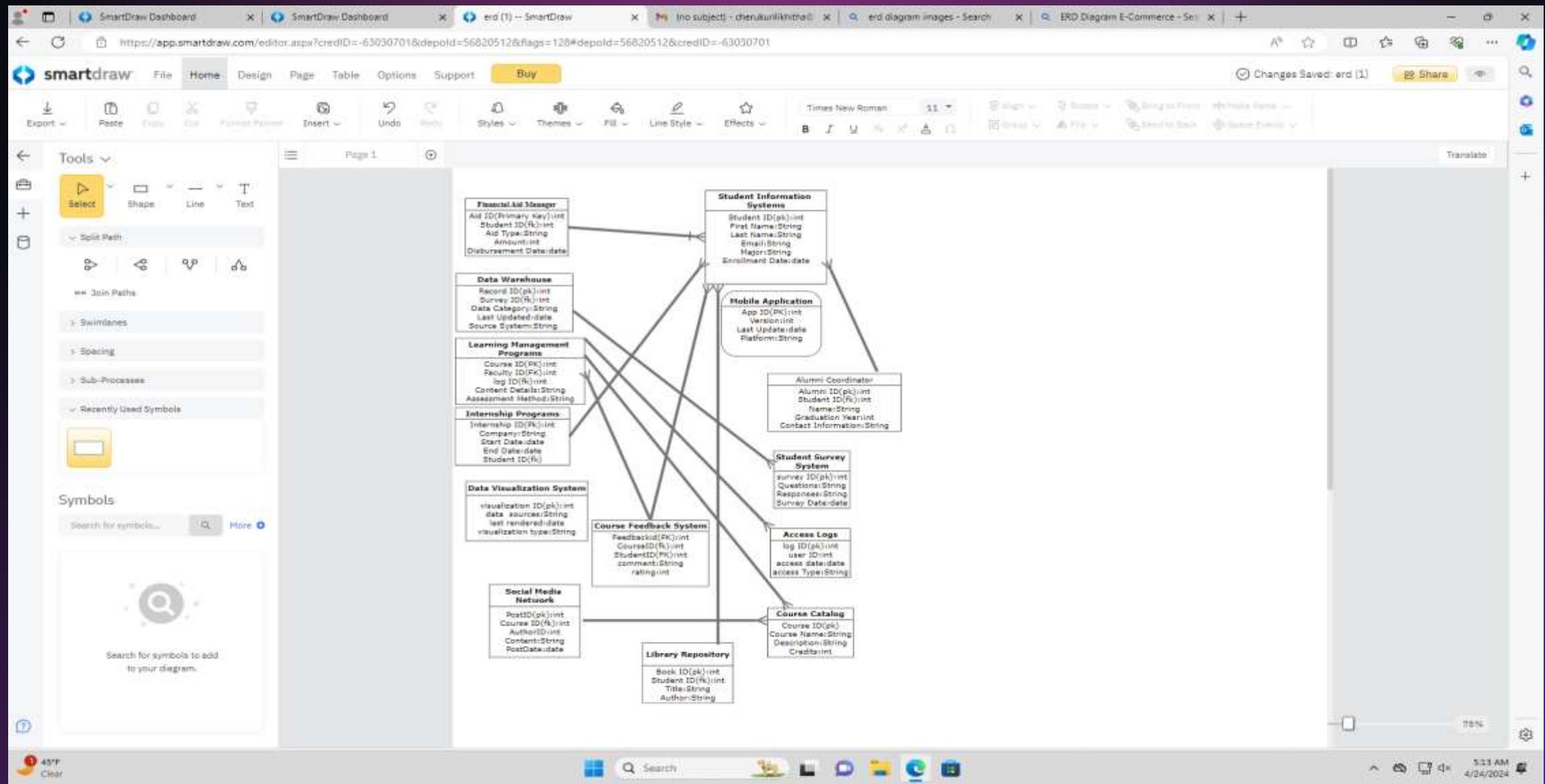
System Taskbar:

- Windows Start
- File Explorer
- Dell Logo
- Google Chrome
- VirusGuard
- WhatsApp
- Microsoft Word
- PowerPoint
- Microsoft Excel
- System Icons (Network, Battery, Volume, etc.)

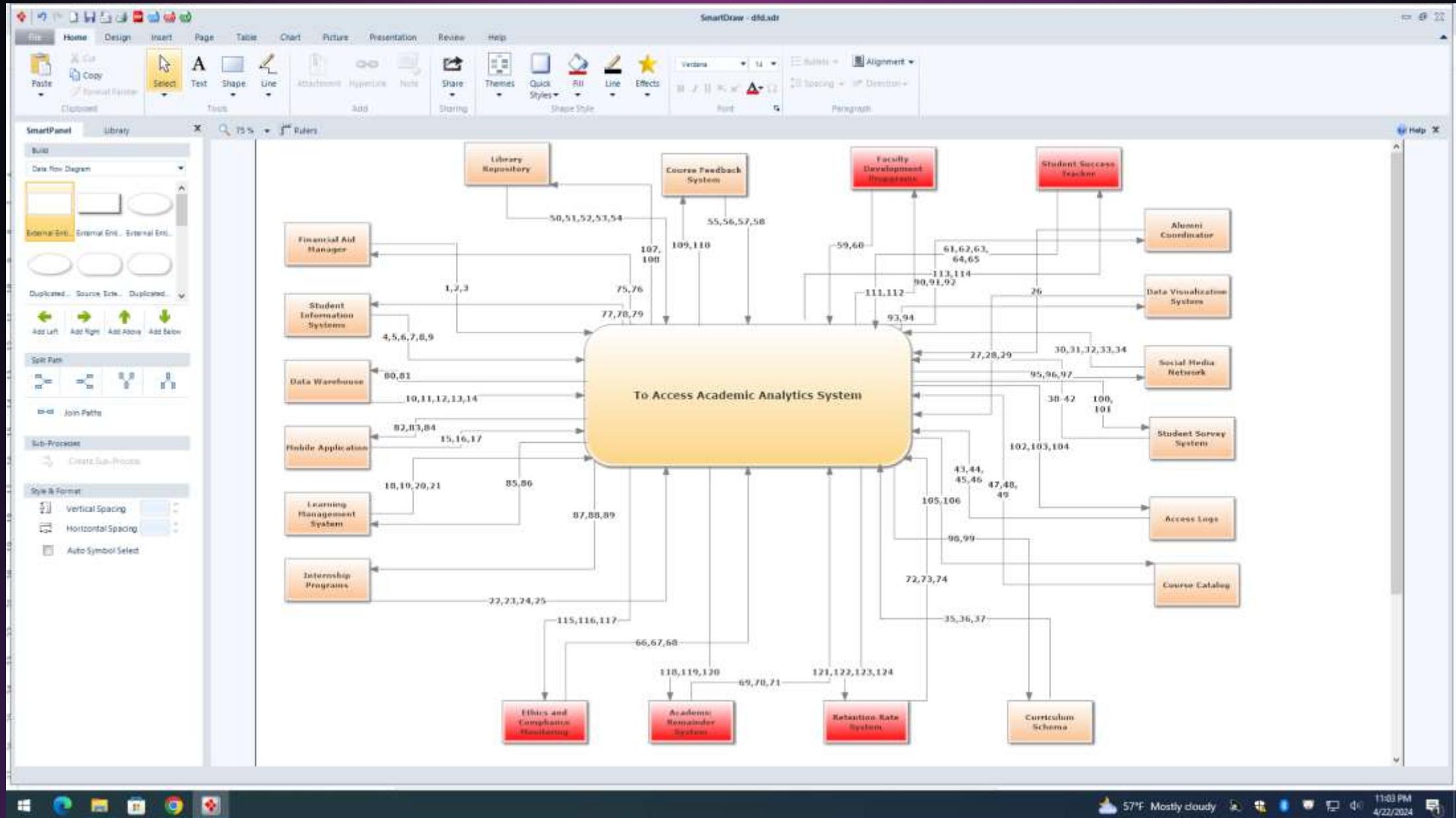
System Status:

- 4:19 PM
- ENG IN
- Wi-Fi
- Battery
- Volume
- 4/20/2024

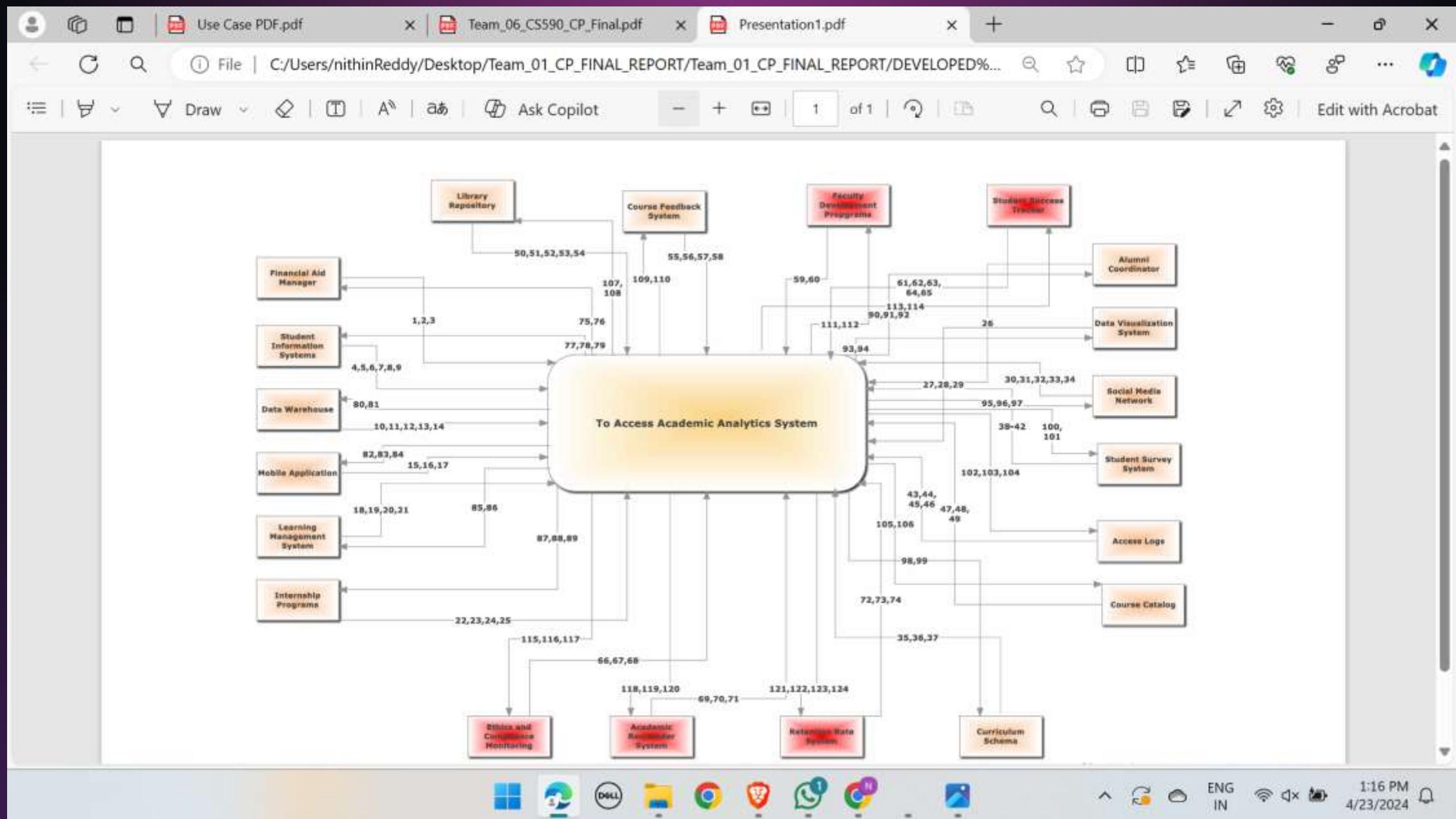
Entity Relationship Diagram Using Smart Draw GUI – Screen Capture



Context Data Flow Diagram Using Smart Draw GUI – Screen Capture



Context Data Flow Diagram Using Smart Draw – PDF



Legends For Context Data Flow Diagram

Financial Aid Manager

1. To Evaluate Eligibility
2. To Disburse Funds
3. To Evaluate Appeal

Student Information Systems

4. Update Student Details
5. Enroll Course
6. Generate Student Report
7. Create Curriculum
8. Edit Curriculum
9. Update Course Schedule
10. Create Warehouse
11. Backup Data
12. Restore Data
13. Monitor Usage
14. Archive Data

Mobile Application

15. Update Version()
16. Update Compatibility()
17. Update Languages()

Learning Management System

18. Add Course()
20. Add Assessment()
21. Calculate Score()
22. Update Program Details()
23. Conduct Training()
24. Monitor Progress()
25. Provide Feedback()

Internship Programs

26. Update Alumni Details

Data visualization tool

27. Create Visualisation()
28. Update Visualisation()
29. Export Visualisation

Social Media Network

30. Create Post
31. Delete Post
32. View Analytics

33. Read Comments
34. Respond Comments

Curriculum Schema

35. Create Curriculum
36. Update Course Schedule
37. To Update Academic Standards

Student Survey System

38. Store Responses
39. Add Comment
40. To provide feedback on courses
41. Identify Highest Common Response
42. Identify Least Common Response

Access Logs

43. Record Access Attempt
44. Retrieve Access Logs
45. Deliver Audit Trails
46. Monitor Access Usage

Course Catalog

47. Enroll Student
48. Check Prerequisites

Legends For Context Data Flow Diagram

49.Manage Waitlists

Library Usage Statistics

50.To Transmit Usage Data

51.To Inform Collection Development

52.To Provide Research Activity Insights

53.To Deliver Citation Analytics

54.To Update Resource Demand Statistics

Course Evaluation Data

55.Store Evaluation()

56.Analyze Results()

57.Notify Instructors()

58.Facilitate feedback()

Faculty Development Programs

59.Add Participant

60.Drop Participant

Student Success Tracker

61.Track Student Progress

62.Update Graduation Status

63.Monitor Grades

64. Generate Progress Status

65.Monitor Performance

Ethics Compliance Monitoring:

66.Add Violation Report()

67.Update Ethics score()

68.Update Compliance score()

Academic Remainder System

69.To provide notification response rates

70.To transmit remainder engagement data

71.To inform user engagement strategies

Retention Rate System

72.To transmit retention statistics

73.To deliver risk analysis

74.To inform policy development

System to Financial aid Manager:

75.To Analyze Eligibility

76.Update Aid Status

System to Student Information Systems:

77.To provide retention data

78.To send progress reports

79.To update achievements records

System to Datawarehouse:

80.To store data

81.To aggregate analytics

System to mobile application:

82.To push notifications

83.To customize user experience

84.To sync dashboard data

System to learning analytics:

85.To enhance learning paths

86.To personalize feedback mechanism

System to internship programs:

87.To assess program alignment

88.To evaluate suitability

89.To recommend candidates

System to Alumni:

90.To supply career outcomes

91. To analyze donation patterns

92.To provide engagement insights

Legends For Context Data Flow Diagram

System to tableau

93.To export visual data

94.To deliver statistical analysis

System to social Media Network:

95.To highlight events

96.To share milestones

97.To promote program updates

System to curriculum schema:

98.To align academic standards

99.To guide course development

System to student survey analysis:

100.Stored responses

101.Analysed the survey

System to access logs:

102. To log data access

103.To monitor user activity

104.To audit compliance

System to course catalog:

105.To update course descriptions

106.To adjust prerequisites

System to library

107. To enhance user services

108. To analyze usage patterns

System to course evaluation data

109.To enhance feedback mechanisms

110.To assess quality assurance

System to faculty:

111.To benchmark teaching performance

112.To guide professional development

System to student success Tracker:

113.To update progress metrics

114.To provide graduation predictions

System to ethics compliance:

115 To report compliance metrics

116.To facilitate audit trails

117.To assess ethical practices

System to academic remainder:

118.To update schedule()

119.To remind enrollment deadlines

120.To co-ordinate event notifications

System to degree:

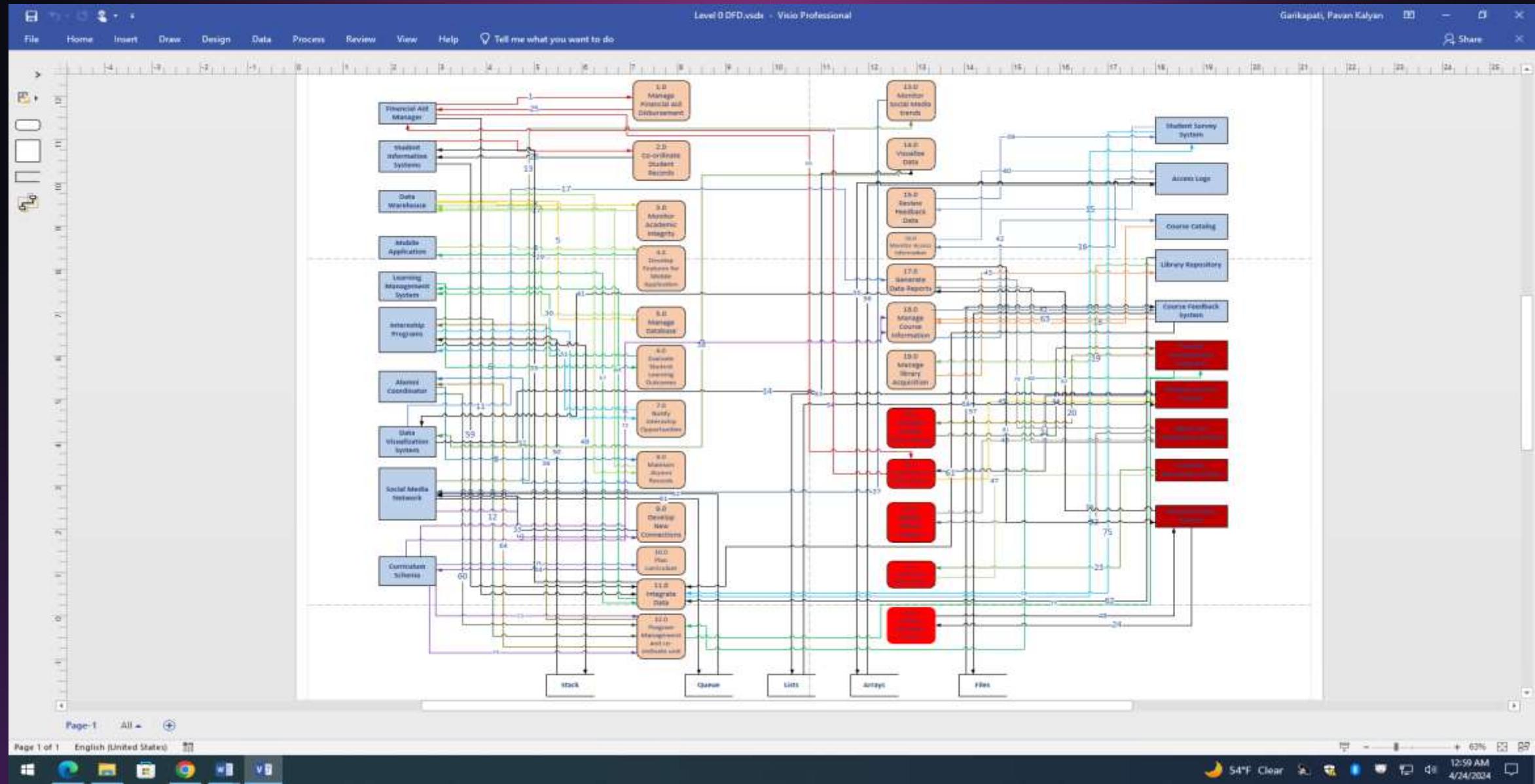
121.To provide retention data

122.To inform intervention programs

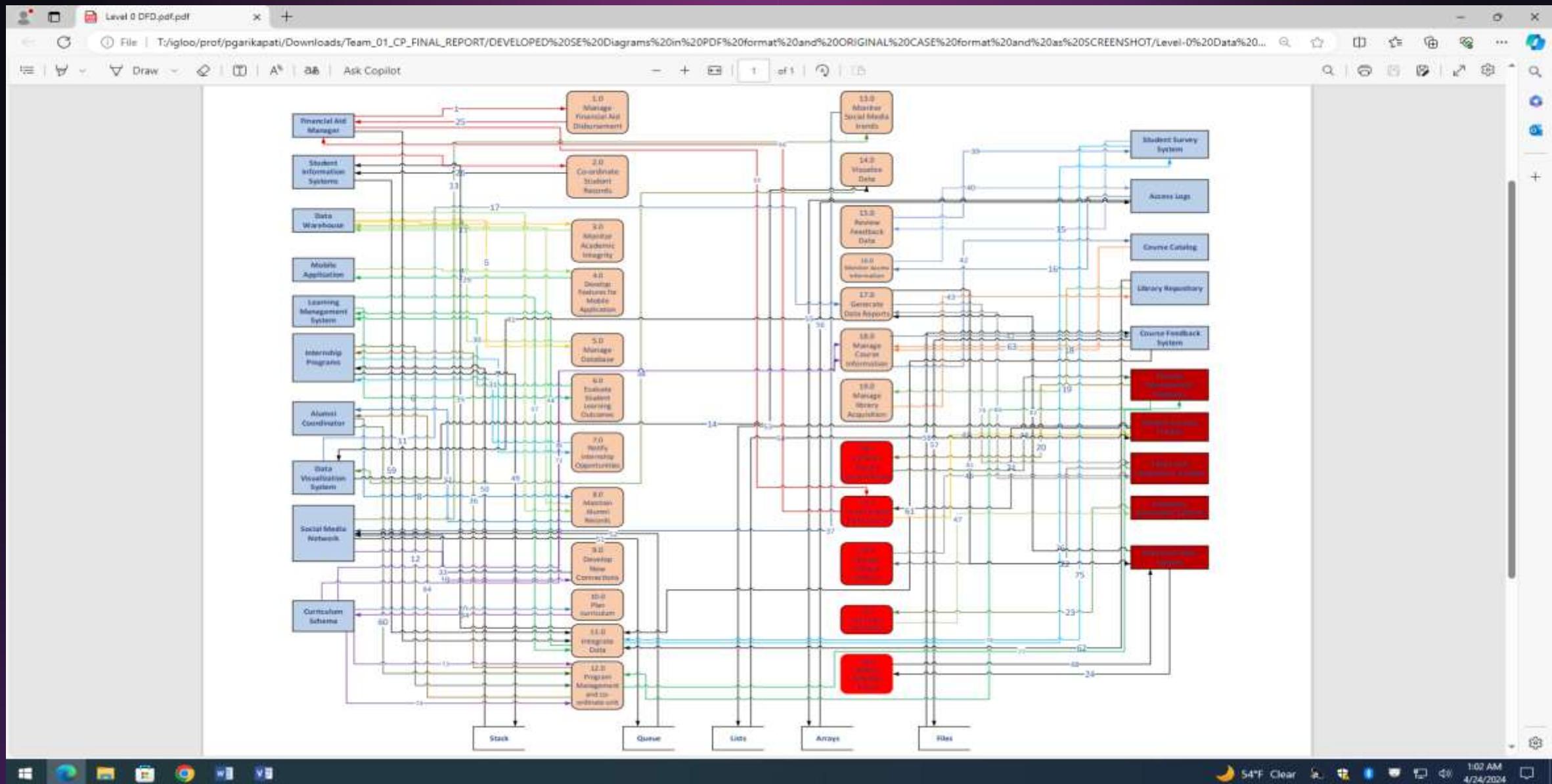
123.To assess program effectiveness

124.To update risk assessments

Level-0 Data Flow Diagram Using VISIO GUI – Screen Capture



Level-0 Data Flow Diagram Using VISIO – PDF



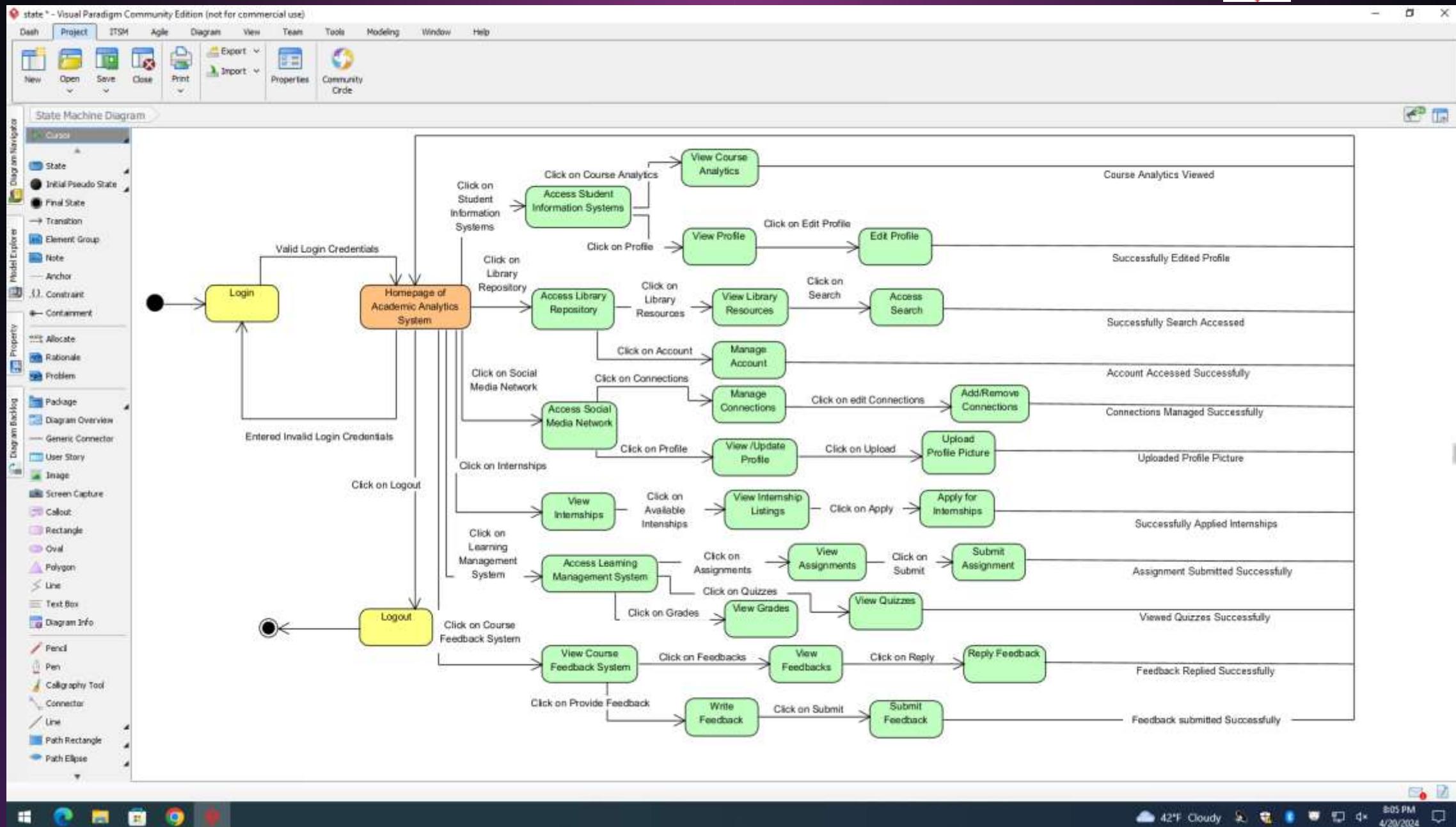
Legends For Level – 0 Data Flow Diagram

- 1. To manage Financial Aid Disbursement
- 2. To Manage Student Records
- 3. To monitor Academic Integrity
- 4. To develop features for Mobile Application
- 5. To manage Database
- 6. To evaluate Student Learning Outcomes
- 7. To notify Internship Opportunities
- 8. To maintain Alumni Records
- 9. To develop new Connections
- 10. To plan Curriculum
- 11. To Integrate Data
- 12. To manage Internship Program
- 13. To monitor social media trends
- 14. To visualize data
- 15. To review Feedback Data
- 16. To monitor Access Information
- 17. To Generate Data Reports
- 18. To Manage Course Information
- 19. To Manage Library Acquisition
- 20. To Evaluate Faculty Performance
- 21. To Track Student Performance
- 22. To Review Ethical Policies
- 23. To Implement Early Alert Systems
- 24. To Analyze Retention Trends
- 25. Updates Financial Information
- 26. Manages Students Records
- 27. Ensures Academic Integrity
- 28. Designs features for Mobile Application
- 29. To maintain Database
- 30. Assess Student Learning Outcomes
- 31. To showcase Internship Opportunities
- 32. Maintains/Updates Alumni Records
- 33. Develops new Connections
- 34. Refines Curriculum
- 35. Integrates Data
- 36. Manages Program Unit
- 37. Analyzes social media trends
- 38. Visualizes data
- 39. Reviews Feedback Data
- 40. Monitors Access Information

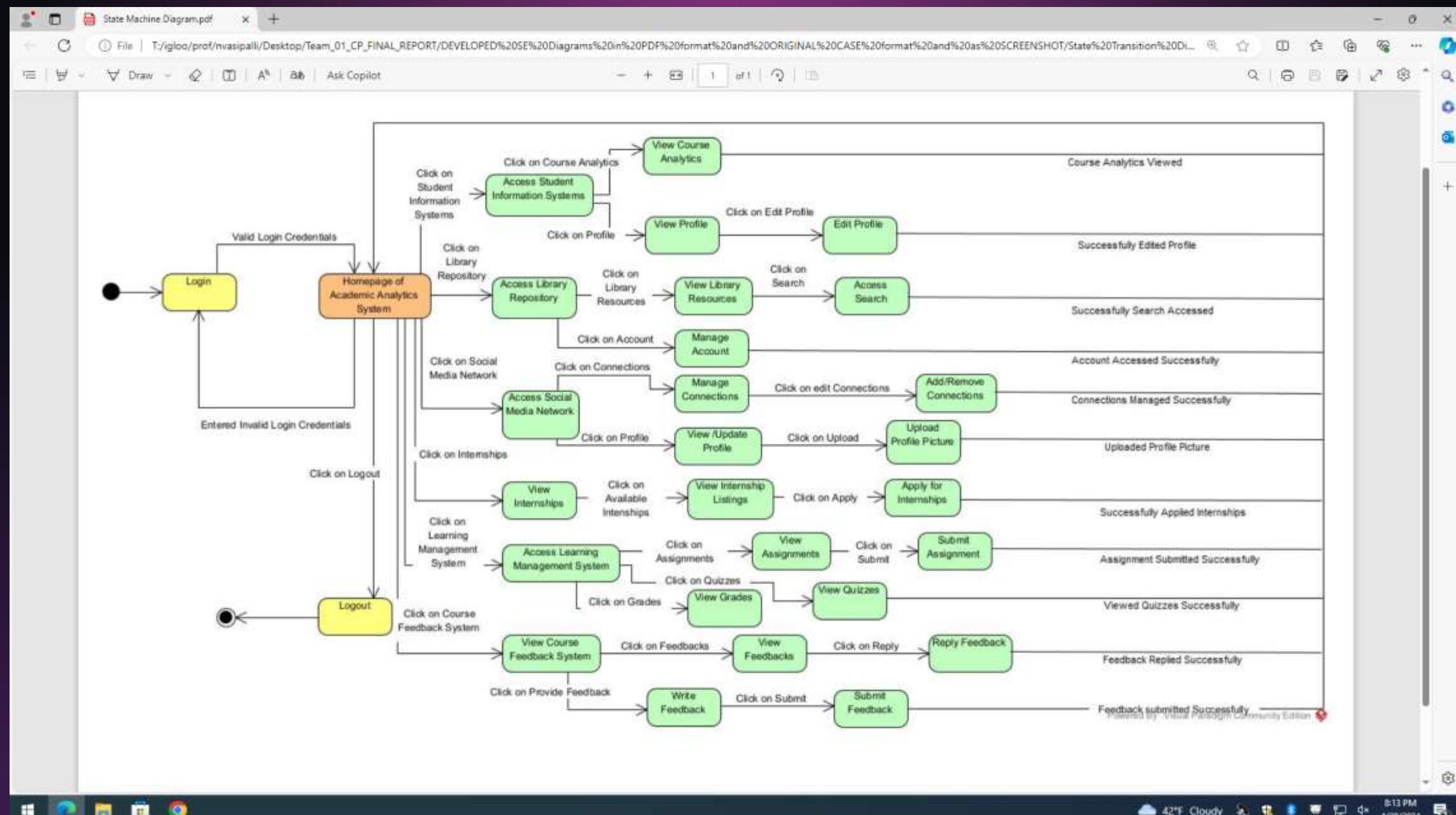
Legends For Level – 0 Data Flow Diagram

- 41.Generates Data Reports
- 42.Maintains Course Information
- 43.Manages Library Acquisition
- 44.Assess Faculty Performance
- 45.Monitor Student Performance
- 46.Updates Ethical Policies
- 47.Implement Early Alert Systems
- 48.To maintain Retention Rates/Trends
- 49. To Retrieve data from Internship Data
- 50. Data Retrieves from Internship Data
- 51. To Retrieve data from Social Media Database
- 52. To maintain data from Social Media Database
- 53.To Store Performance in Lists
- 54.To maintain Performances in Lists
- 55.To Preserve Log Information into Arrays
- 56.To maintain Log Information into Arrays
- 57. To Store evaluation report in files
- 58. To manage evaluation report in files
- 59.To Integrate student records
- 60.To manage Programs for Alumni Co-ordination
- 61.To Integrate Course Feedback Information
- 62.To Integrate Library Acquisition
- 63.To Manage Course Feedback data
- 64.to maintain Program for Alumni Co-ordination
- 65.To track student Performance for financial manger
- 66.To view the student performance
- 67.To integrate learning data
- 68.To process the data
- 69.To view social media report
- 70.To generate social media report
- 71.To manage course information for curriculum schema
- 72.To maintain the course information
- 73.To manage curriculum schema planning
- 74.To maintain curriculum schema program
- 75.To integrate student survey data
- 76.To process student survey data
- 77.To manage the faculty development programs
- 78.To maintain the faculty development programs
- 79.To view ethics and compliance data programs
- 80.To generate ethics and compliance data reports
- 81.To view retention data reports
- 82.To generate retention data reports

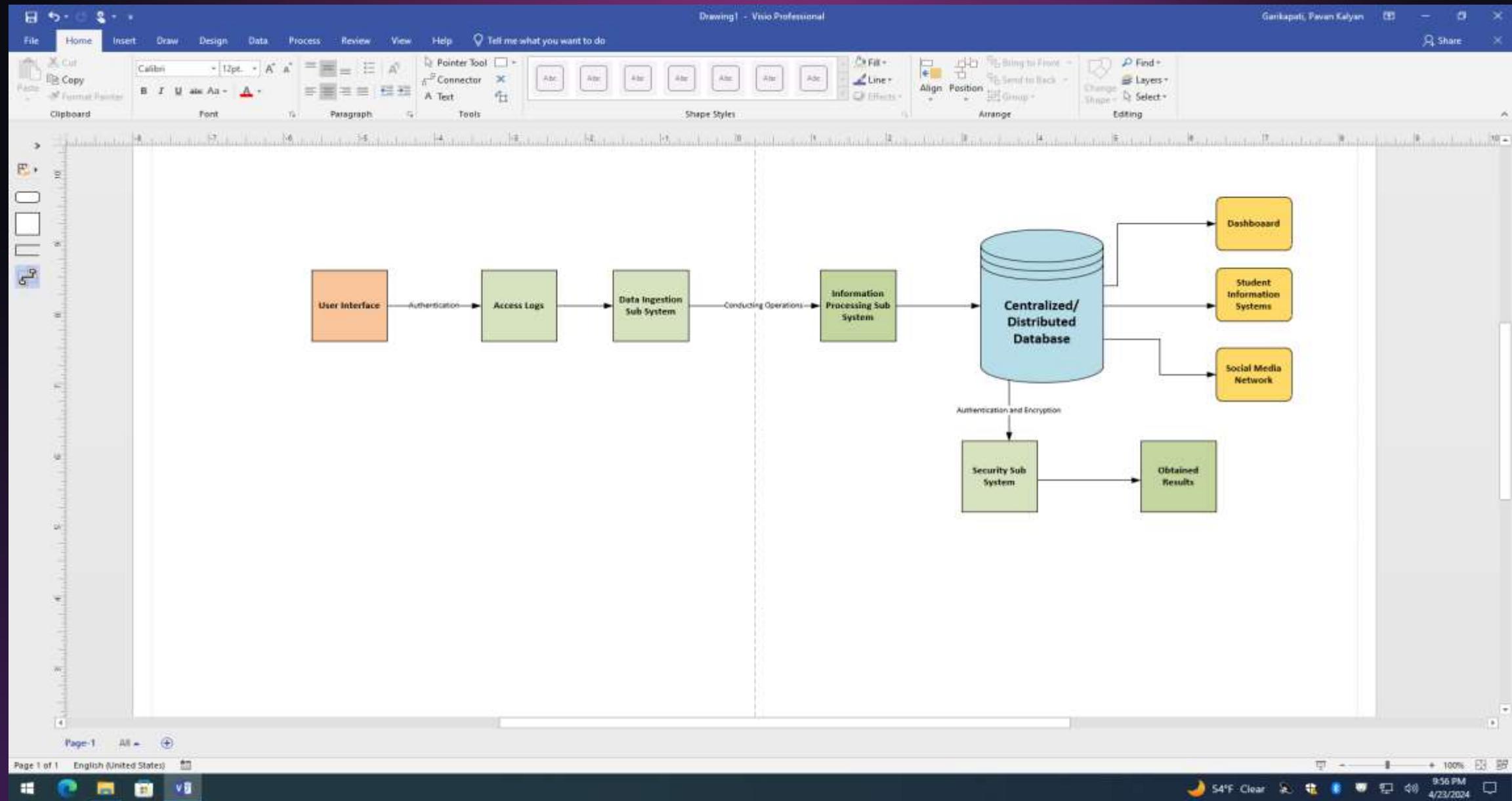
State Transition Diagram Using Visual Paradigm GUI – Screen capture



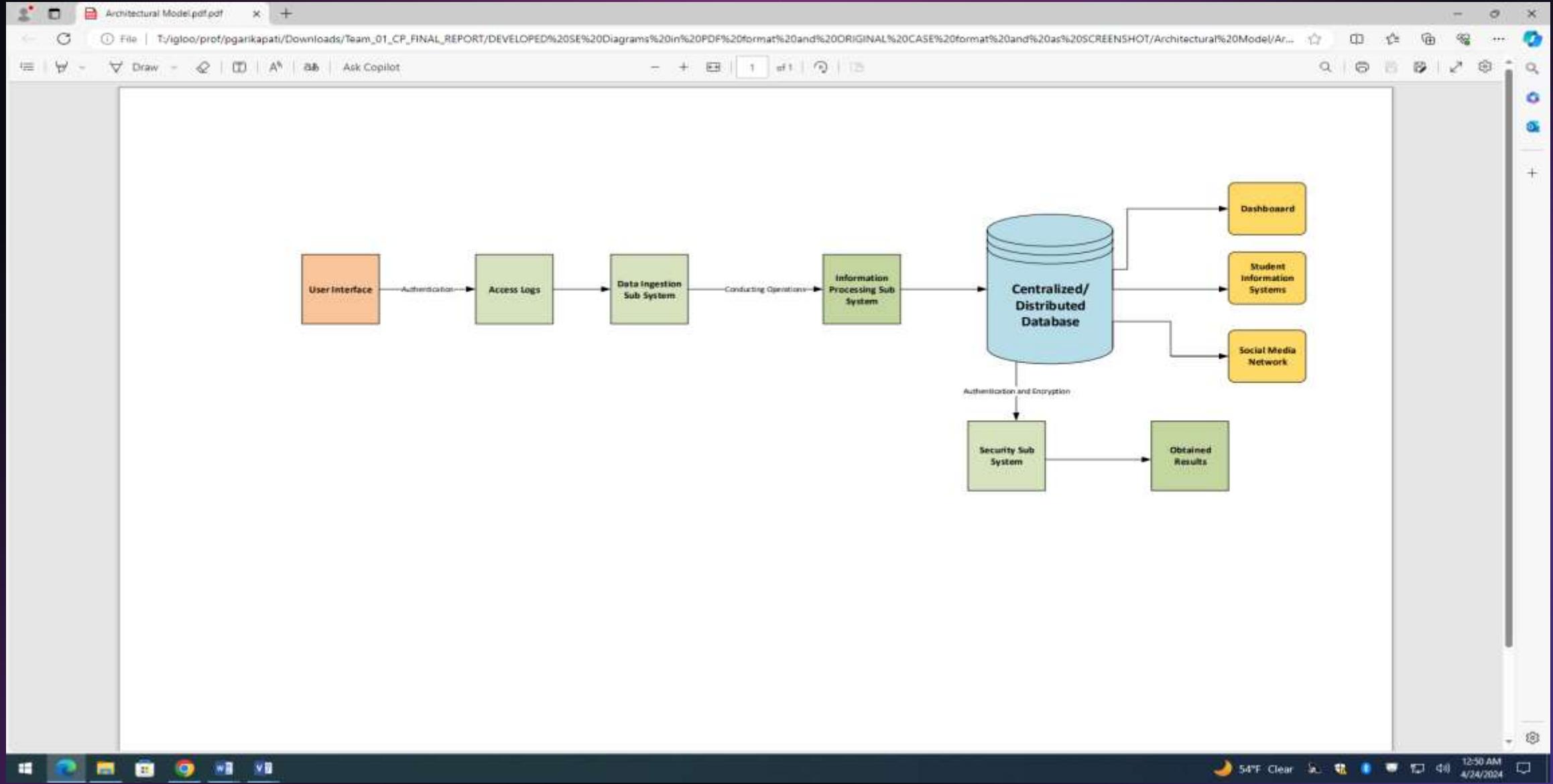
State Transition Diagram Using Visual Paradigm – PDF



Software Architectural Model Using Visio GUI – Screen Capture



Software Architectural Model Using Visio - PDF



This software architectural diagram represents the flow and storage of data for an academic analytics system. Here's a description of main components and their interactions :

User Interface (UI):

This is where users, such as students, faculty, or administrators, interact with the system. Through the UI, they can input data, run queries, and request reports. It's the front-end component that users see and interact with.

Data Ingestion Sub System:

The Data Ingestion Subsystem is responsible for collecting, processing, and integrating data from various sources into the Academic Analytics System. It serves as the entry point for raw data, transforming it into a format suitable for analysis and decision-making.

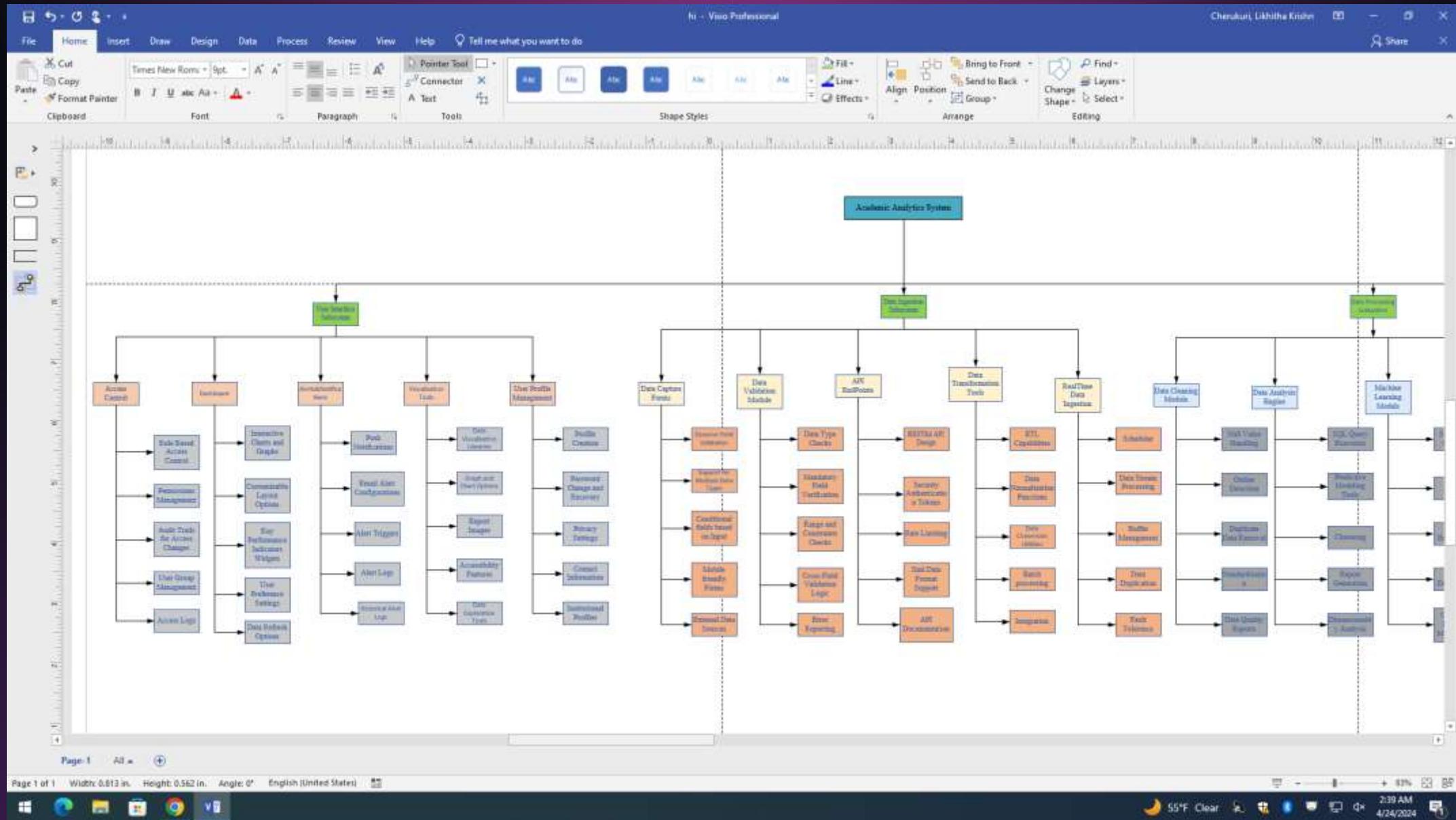
Information Processing System:

Once the user's request is authenticated and logged, it is sent to an information processing system. This component handles the business logic of the application, such as calculating analytics, processing academic data, and preparing information for storage or presentation.

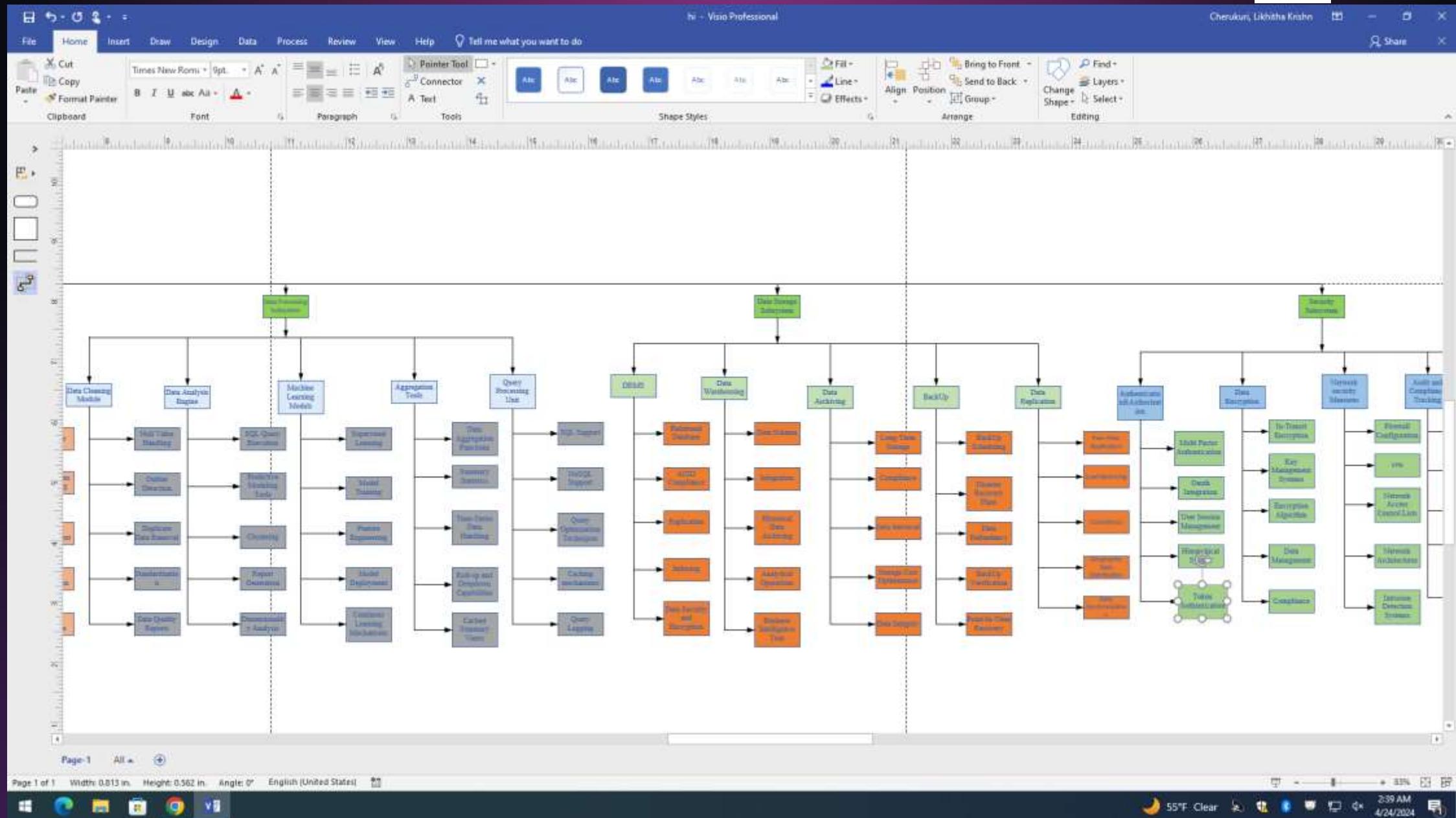
Centralized/Distributed Database: This is the data storage component of the architecture, where all academic data is stored. Depending on the design, the database can be a single central repository (centralized) or spread across multiple locations (distributed), which can help with load balancing and ensuring data availability.

Outputs: The system outputs data in various forms: Dashboard, Student Information Systems, Social Media Network, and Obtained Results.

Software System Hierarchical Design Model Using Visio GUI – Screen Capture



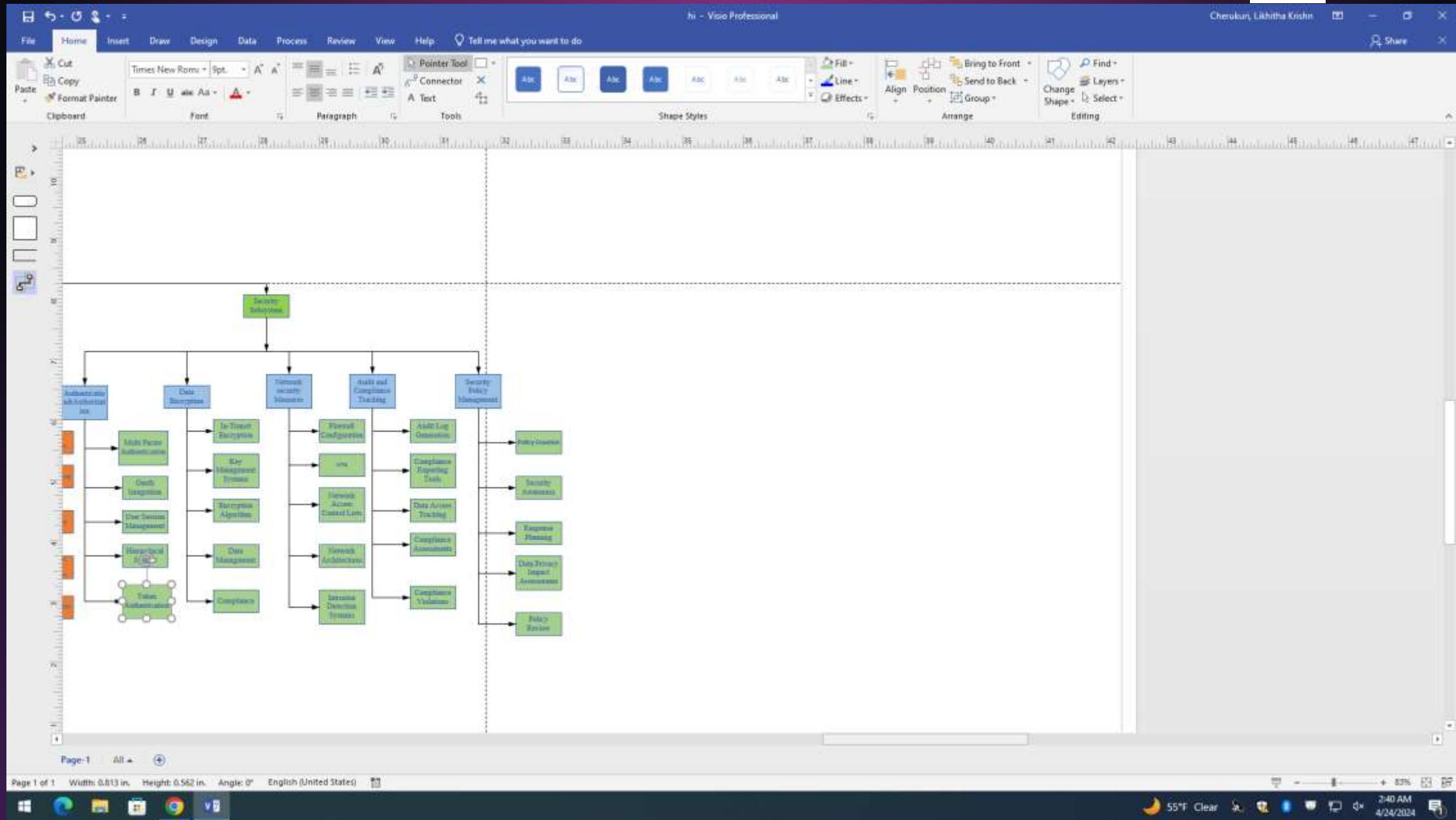
Software System Hierarchical Design Model Using Visio GUI – Screen Capture



Software System Hierarchical Design Model Using Visio GUI – Screen Capture

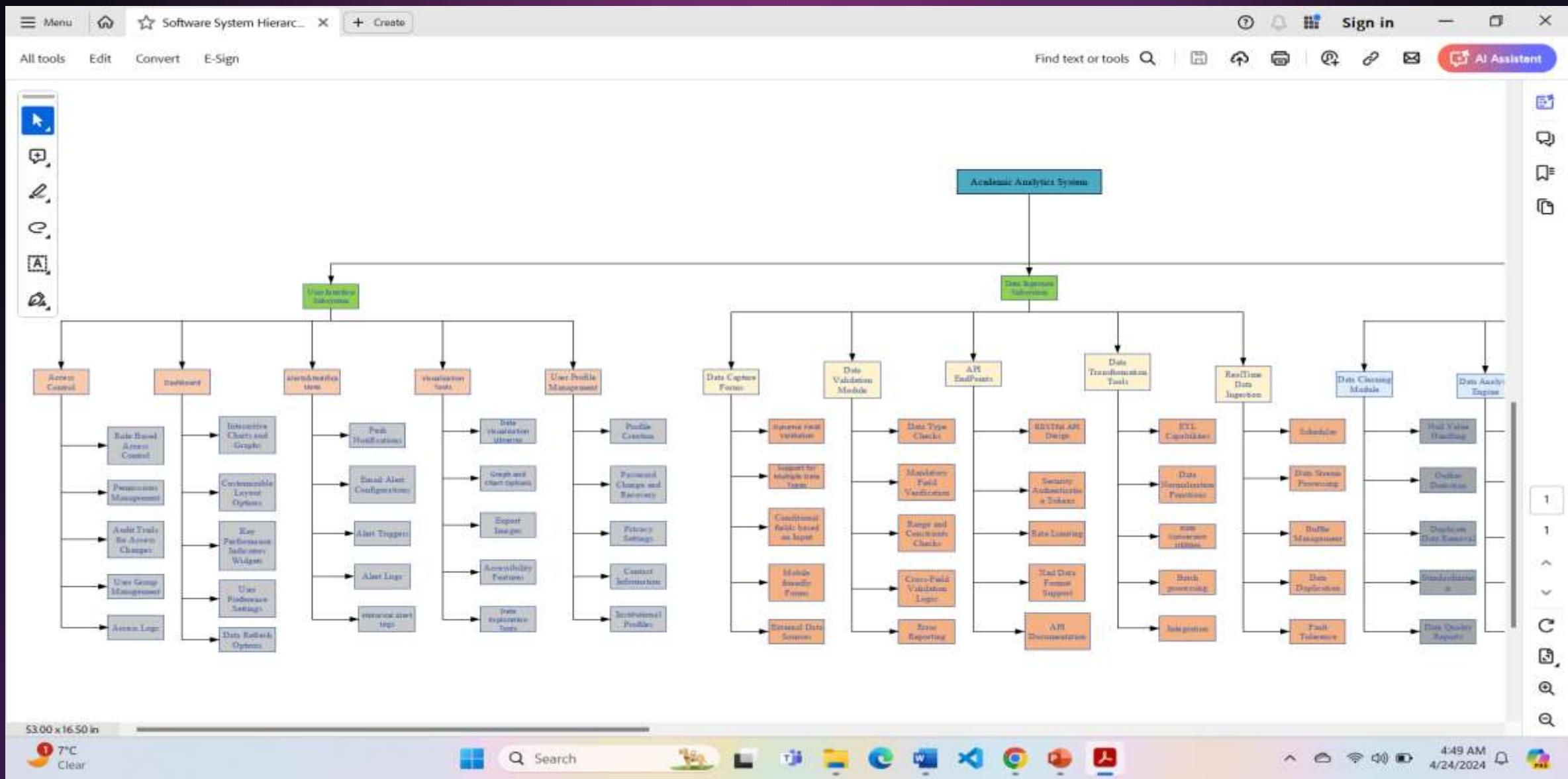


42



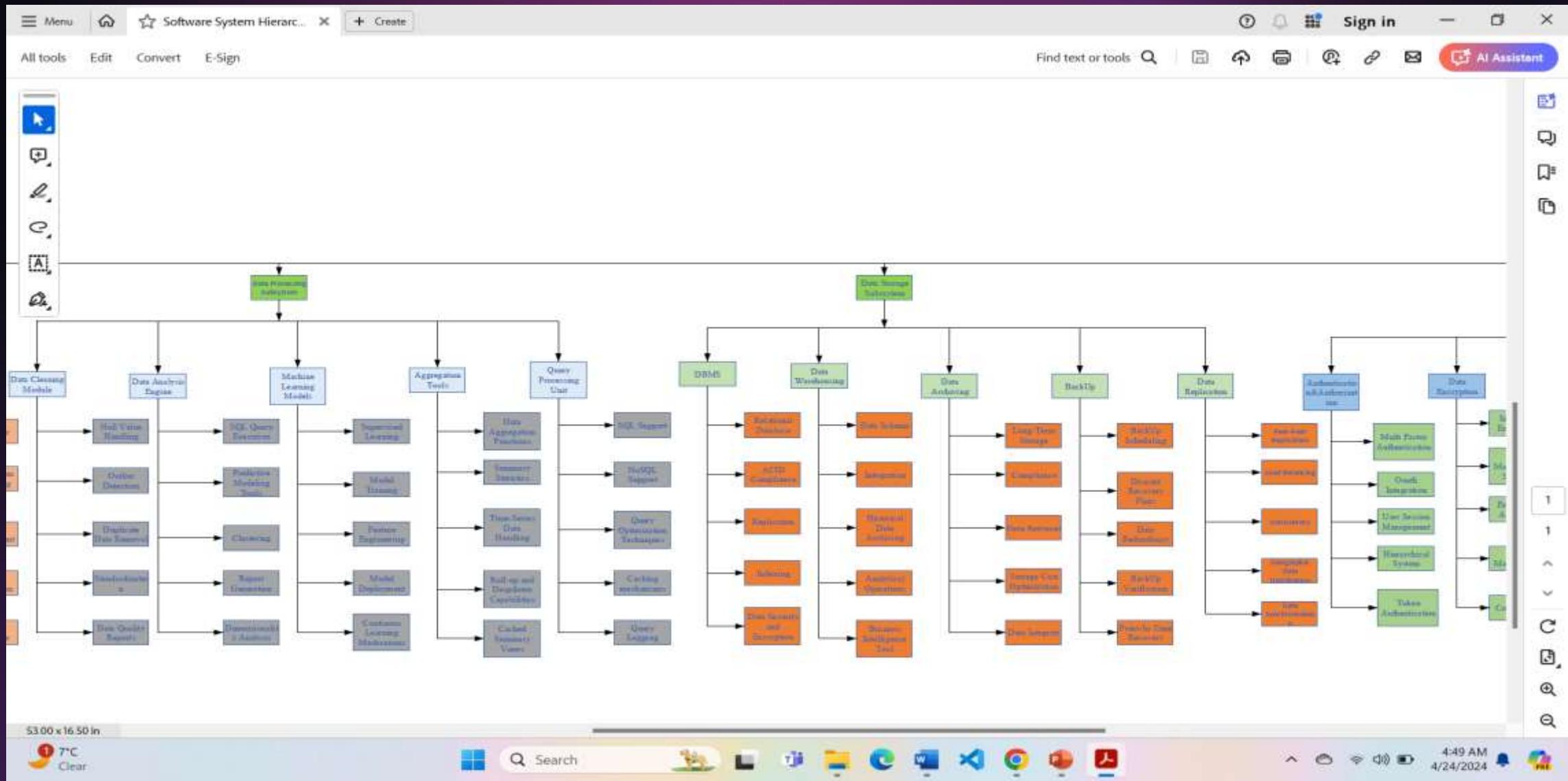


Software System Hierarchical Design Model Using Visio - PDF



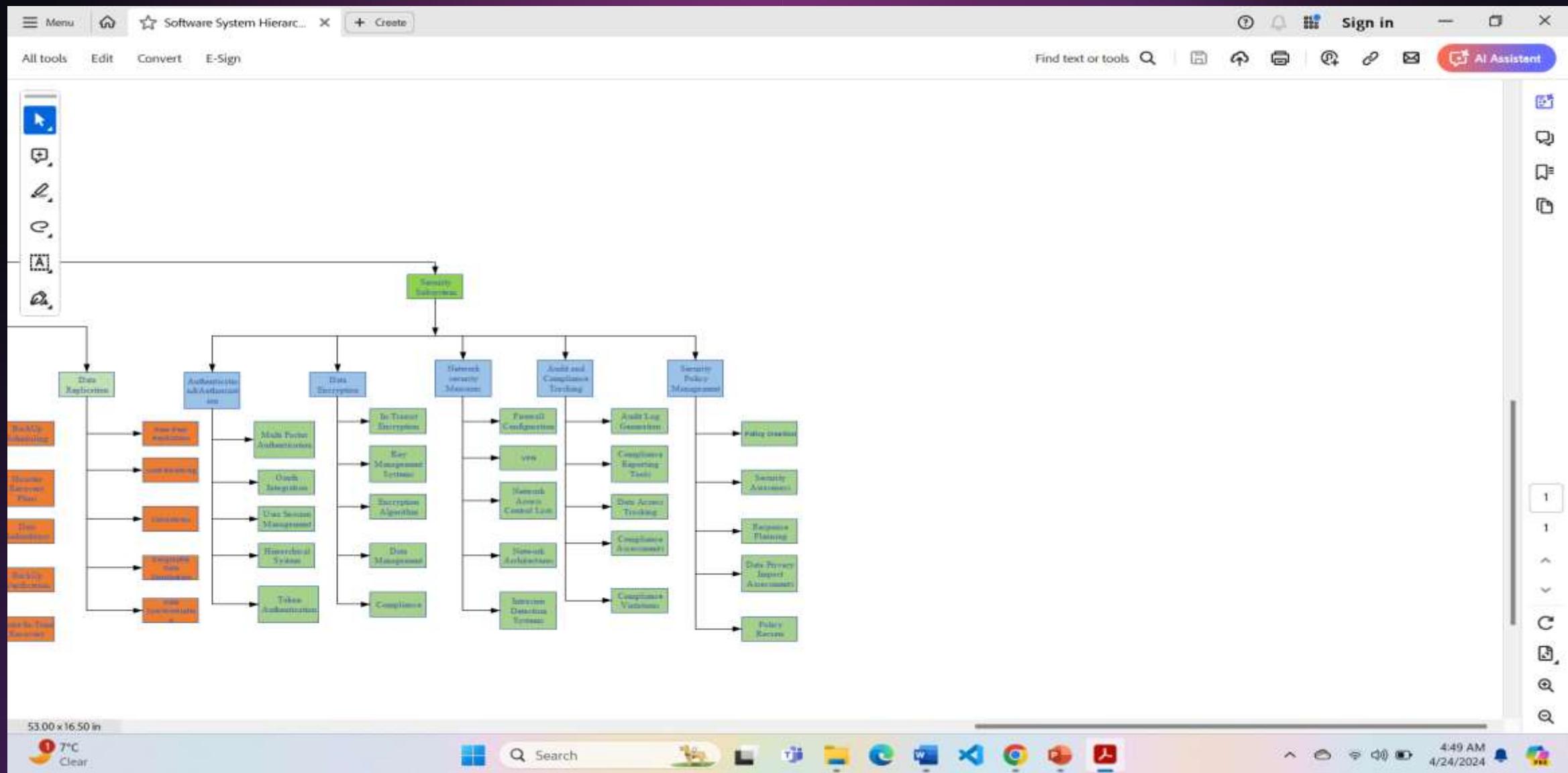


Software System Hierarchical Design Model Using Visio- PDF

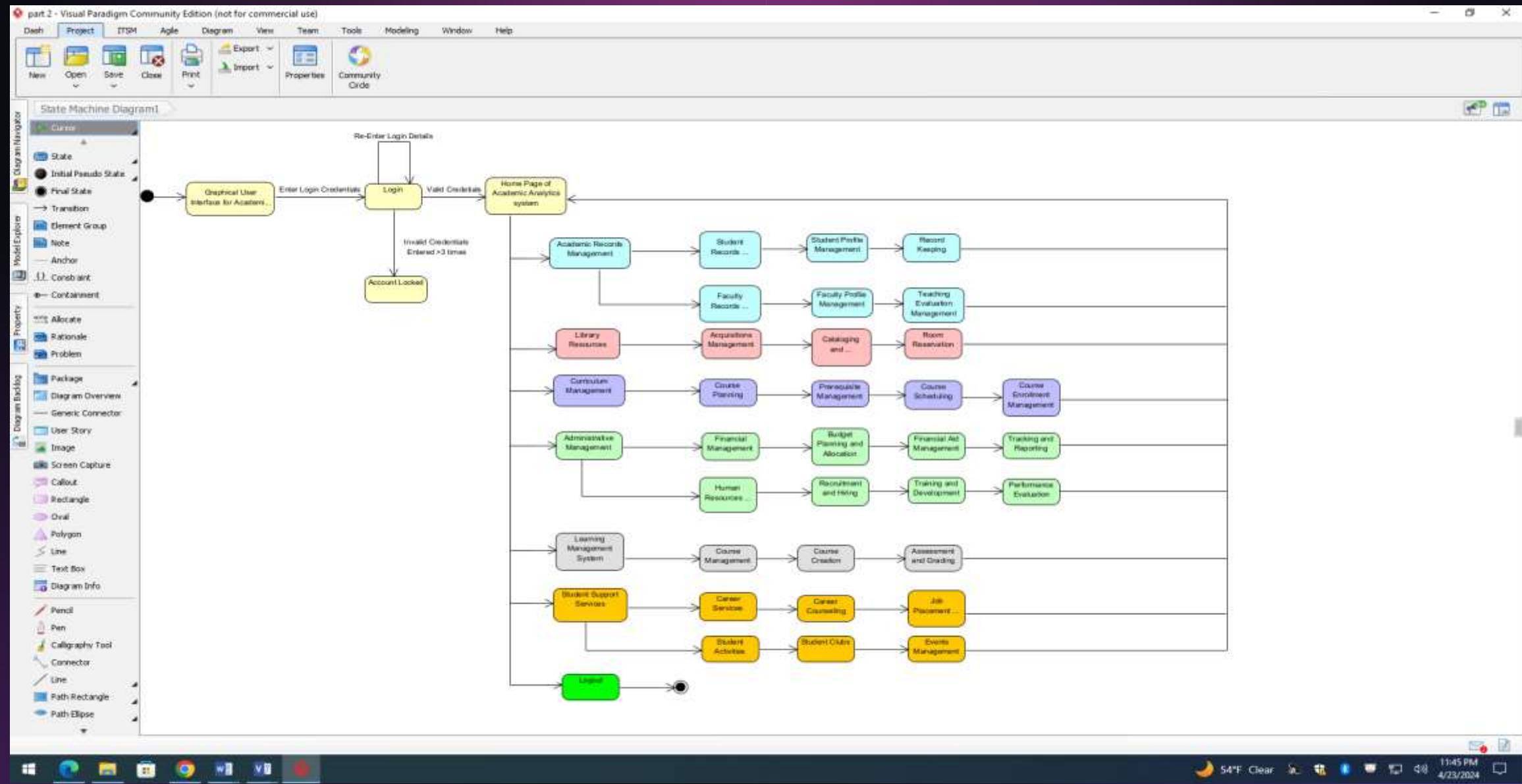




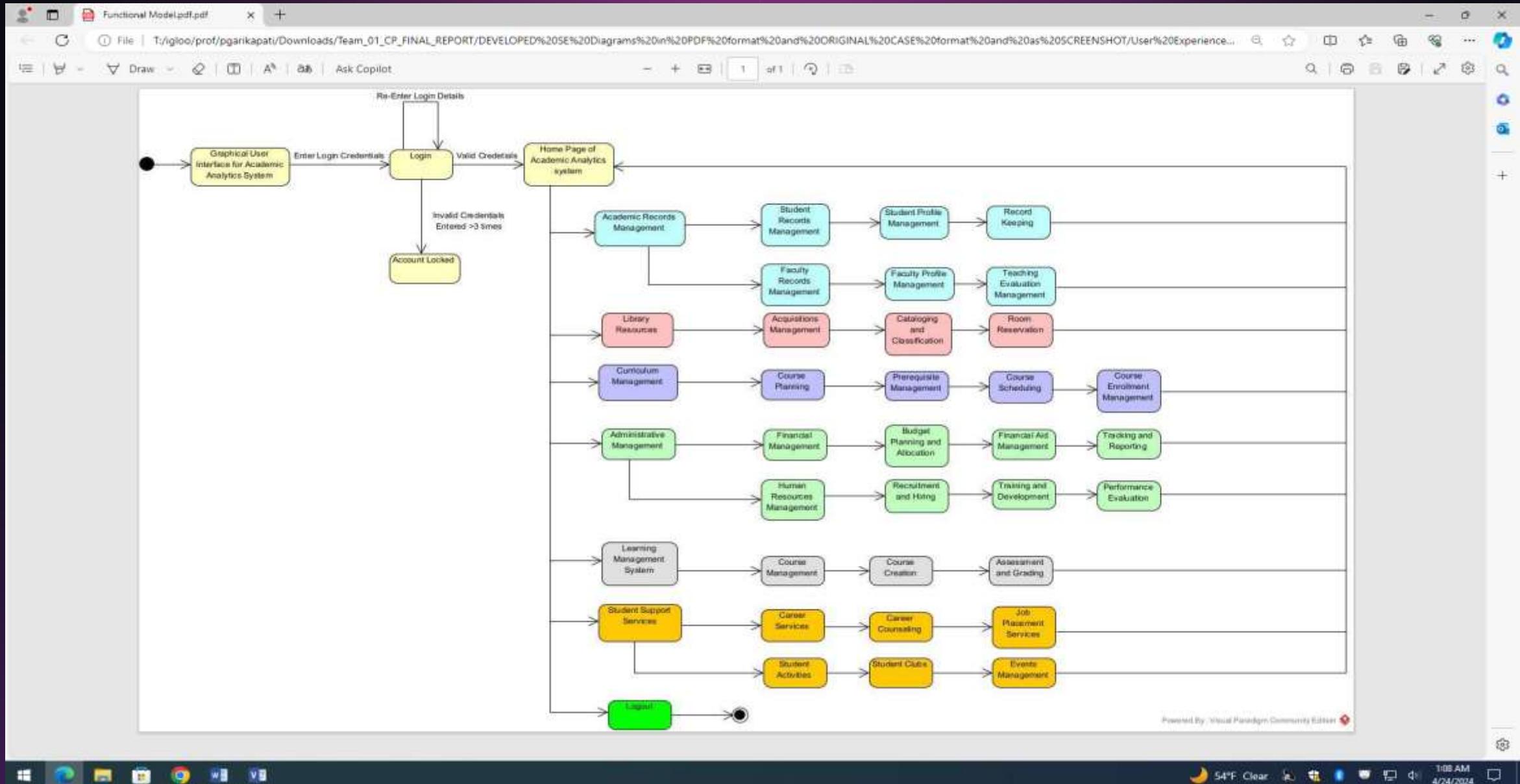
Software System Hierarchical Design Model Using Visio - PDF



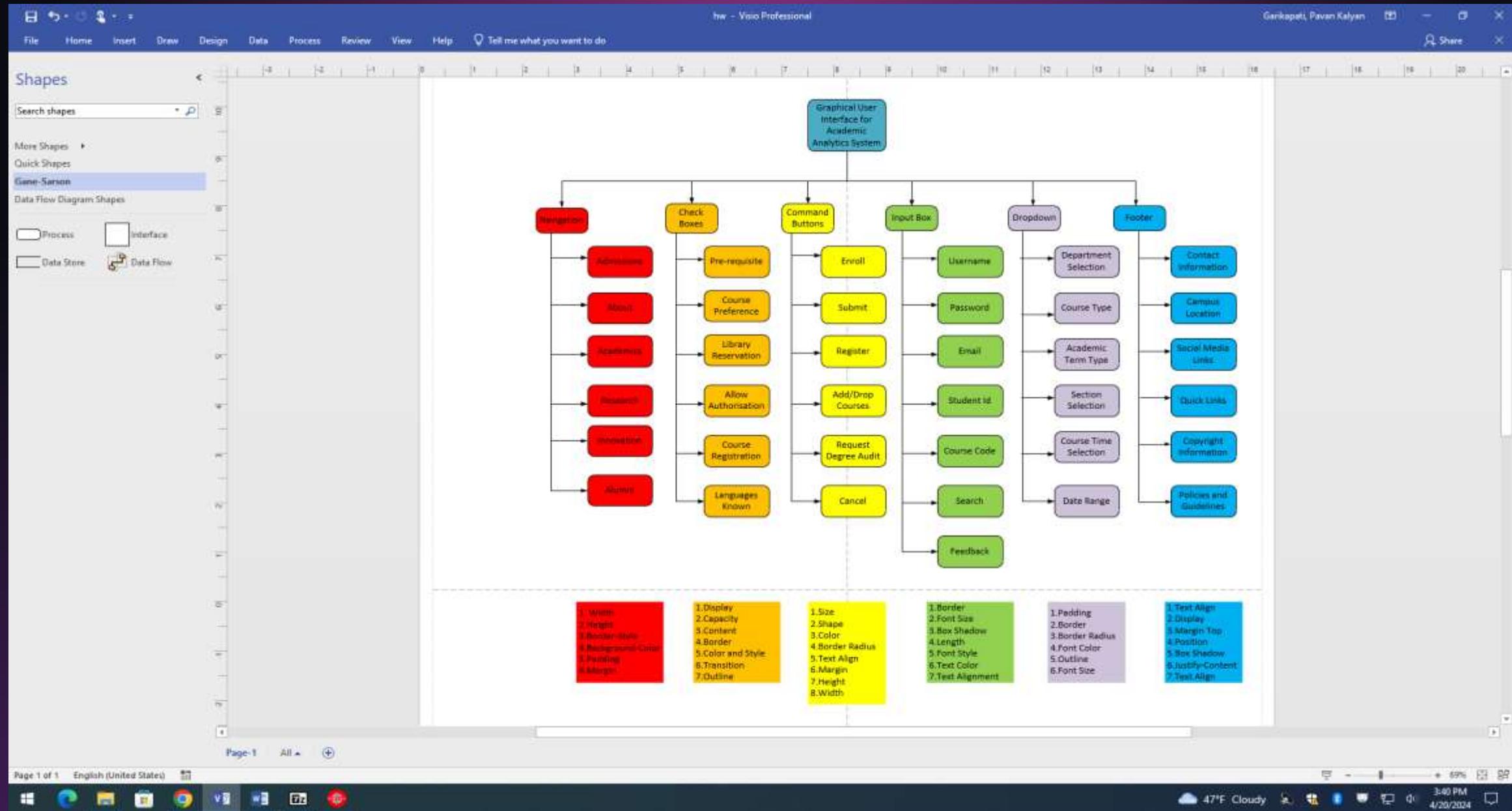
UE's Hierarchical Design – Functional Model GUI - Screen Capture



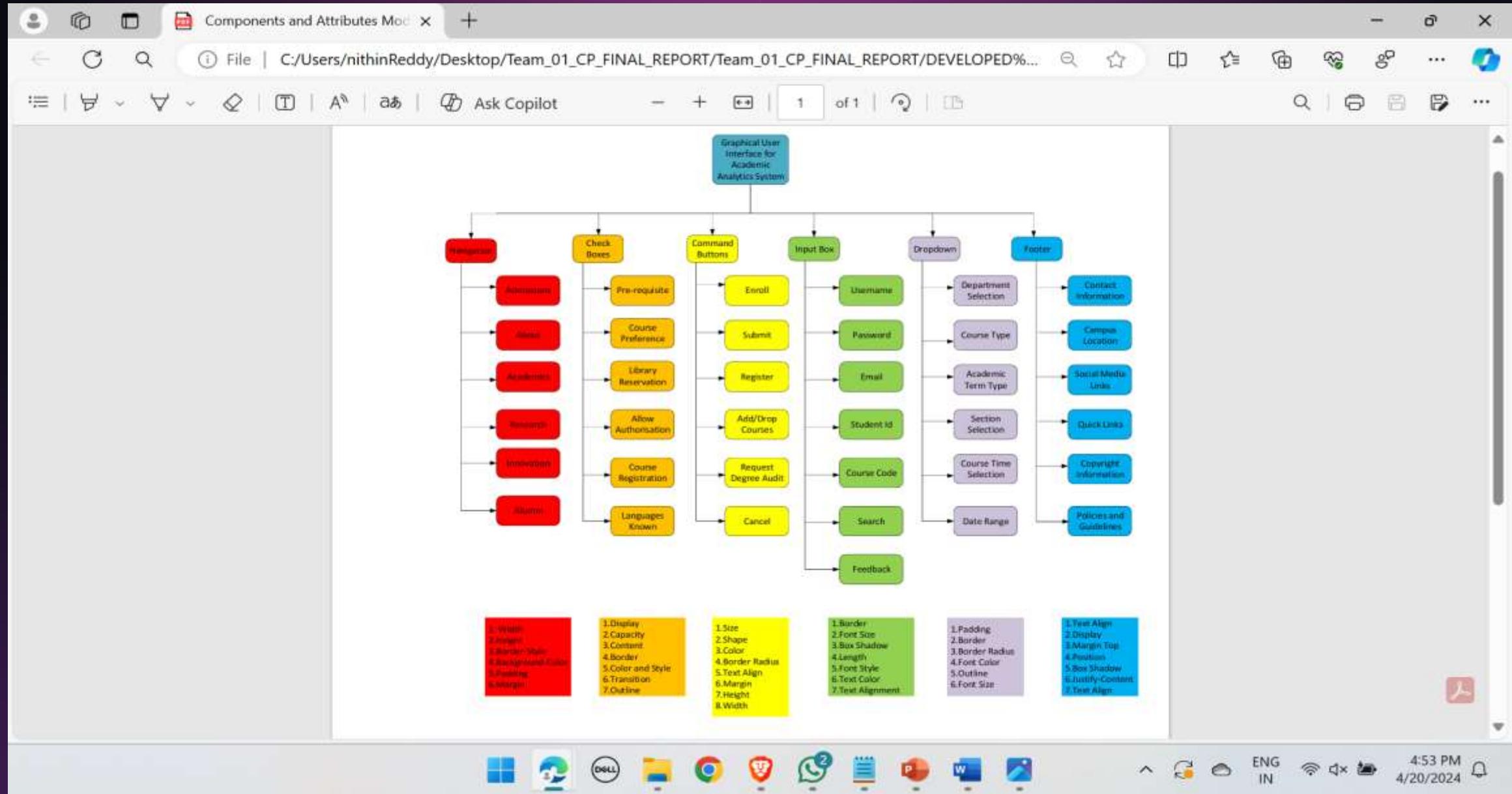
UE's Hierarchical Design – Functional Model – PDF



UE's Hierarchical Design - Components and Design Model GUI - Screen Capture



UE's Hierarchical Design - Components and Design Model - PDF



LOC Based Project Cost Estimation

System Function	Programming Language	Estimated LOC
Database management system	PHP MySQL	20,000
Computer Graphics Display Functions	Javascript	55,000
User Experience Interface	HTML,CSS,XML	15,025
Programming Languages	Java	59,928
Data exchange protocols, security and communication between main sub-systems	Auth(),javascript	12,000
Control of Peripherals	Js with web API's	5,000
Total		1,66,953

- Estimated Lines Of Code (LOC) = 1,66,953
- Average cost per line of code as per market rate = \$10
- Burdened labor rate= \$8000 per month
- Estimated project cost = (Estimated LOC * Average cost per LOC) = $(1,66,953 * \$10) = \$16,69,530$
- Total Estimated Project Effort (in person-months) = $\$16,69,530/8,000 = 208$ person-months = 17 person years

FP Based Project Cost Estimation

System Function	Optimal	Most Likely	Pessimistic Estimate	Estimate Count	Weight	Function - Point Count
Number of inputs from the environment to the system	18	20	25	21	5	105
Number of outputs from the system to the environment	15	18	20	17	6	102
Number of inquiries	65	70	72	69	9	690
Number of internal data structures	75	80	70	75	10	750
Number of external interfaces to various systems and databases	25	27	30	27	8	216
Total						1863

- Cost per FP = 8,000
- Based on LOC estimate the total estimated project cost:
- Total FP count * cost/FP = $1863 * 8000 = \$14,904,000$
- The productivity rate according to the market rate is 10
- The estimated effort is total FP count/Productivity rate = $1863/10 = 186$ person months
 $= 15$ person years

Conclusion

- The Academic Analytics System has been designed to fundamentally transform the way educational institutions manage and leverage data to enhance decision-making processes and academic outcomes. Throughout the development of this system, we adhered to robust software engineering principles to ensure reliability, scalability, and user-friendliness.
- Key achievements of the Academic Analytics System include the integration of diverse data sources which provides a holistic view of academic performance and operational efficiency. This system empowers administrators and faculty by providing actionable insights through advanced analytics and visual dashboards. Additionally, the modular architecture allows for future enhancements and integration with emerging technologies.
- However, as with any system, continuous improvement is critical. Future versions of the Academic Analytics System should focus on incorporating machine learning algorithms to predict student outcomes more accurately and identifying areas for institutional improvement. Moreover, expanding the system's capabilities to include mobile platforms would increase accessibility, allowing for real-time decisions based on the latest data.
- In conclusion, the Academic Analytics System stands as a testament to the power of modern software engineering applied within the educational sector. It not only optimizes current processes but also paves the way for data-driven strategies that promise to elevate the standards of education. The continued evolution of this system will undoubtedly play a pivotal role in shaping the educational landscapes of the future.

THANK YOU