

# SOFTWARE ENGINEERING & CONCEPT

## PROJECT

### RESULT ANALYSIS SYSTEM

NAME: SHANMUGA DIVYA K

REGISTER NO:220701261

DEPARTMENT:COMPUTER SCIENCE & ENGINEERING

YEAR:II-D

SUBJECT CODE:CS19442

# TABLE OF CONTENT:

<b>RESULT ANALYSIS SYSTEM .....</b>	<b>1</b>
Business Need of the Project .....	7
Current Process .....	7
Manual Process .....	7
Automated Process .....	8
Personas and Their Involvement .....	8
Teachers/Instructors .....	8
Administrators .....	8
Students .....	8
Business Problems .....	8
Inefficient Data Management .....	8
Limited Analytics .....	9
Poor Visibility .....	9
Data Silos .....	9
Conclusion .....	9
Architecture Diagram .....	12
Class Diagrams .....	13
Sequence Diagrams .....	14
Test Strategy .....	14
Test Cases for User Stories .....	15
GitHub Repository Structure .....	16
DevOps Architecture in Azure .....	16
Components .....	17
Deployment Strategy .....	18
Security Considerations .....	18

## Overview of the Project:

The Project Performance Analysis System serves as a critical tool for organizations to comprehensively evaluate their outcomes and performance of their projects. It addresses several common challenges faced in project management, offering a structured approach to assessing project success and failure.

One primary problem it resolves is the lack of standardized evaluation methods and fragmented data sources. Often, organizations struggle to gather relevant project data from disparate sources, leading to inconsistencies in analysis and decision-making. The system tackles this issue by integrating data from various project-related sources, including plans, budgets, timelings, and performance metrics. This ensures that stakeholders have access to a centralized repository of project data, eliminating the need for manually gathering information from multiple sources.

From a data perspective, the system employs advanced analytics techniques to interpret large volumes of data and derive actionable insights regarding project performance. Data is standardized and structured to ensure consistency and accuracy, enabling meaningful analysis and comparison across different projects. Visualization tools such as charts, graphs, and dashboards are utilized to present analysis results in an intuitive and digestible format, facilitating better understanding and decision-making.

The implementation of the Project Performance Analysis System brings several benefits to users within the organization. Firstly, it enables conducting comprehensive performance evaluations, providing a holistic view of project success or failure based on predefined criteria and metrics. This empowers stakeholders to identify areas of improvement, replicate successful practices, and learn from past mistakes. Moreover, the system facilitates informed decision-making by providing timely access to accurate and comprehensive project data. It helps in evaluating the performance of projects against industry standards, driving continuous improvement through initial reviews and mitigating risks effectively. Optimizing processes, managing risks, and improving overall project outcomes are achieved through thorough analysis and informed decision-making.

## Business Architecture Diagram:

The Business Architecture Diagram for the Project Portfolio Analysis System depicts the structural framework and functional components of the system within the organizational context. It's clear, the diagram illustrates how the system aligns with the organization's overarching goals and objectives, serving as a catalyst for improved project management practices and decision-making processes.

The diagram showcases the integrated relationships between various stakeholders, including project managers, team members, stakeholders, and external bodies. It highlights how these stakeholders interact with the system, from input and analysis to decision-making and feedback loops.

Key functional components of the system are depicted, including data collection, analysis tools, visualization dashboards, and reporting mechanisms. These components work synergistically to support project portfolio analysis processes, enhancing data-driven decision-making capabilities and facilitating knowledge sharing and collaboration across the organization.

Moreover, the diagram illustrates the flow of information within the system, emphasizing the importance of data quality, security, and accessibility. It underscores the role of the system in centralizing project-related data, standardizing analysis methodologies, and promoting transparency and accountability throughout the project lifecycle. The core of the Project Portfolio Analysis Software consists of various modules designed to analyze and integrate different types of data, ensuring a robust and user-friendly platform for analyzing and managing various aspects of the organization's projects.

Once the design is finalized, the development phase begins with a comprehensive front-end and back-end development, implementation, and integration of analysis algorithms. Throughout this phase, rigorous testing procedures are employed to guarantee the reliability and functionality of the software.

Upon successful testing, the software undergoes deployment, with the background deployment to a suitable server environment and the front-end distribution to end-users. Post-deployment, the project's general maintenance and support phases, including ongoing updates, bug fixes, and technical assistance, are provided to ensure smooth operation and user satisfaction.

This comprehensive approach guarantees the highest quality solution by leveraging the strengths of its users while adhering to industry standards and best practices.

Overall, the Business Architecture Diagram provides a holistic view of how the Project's Project Portfolio System integrates with the organization's business processes and objectives, ultimately driving efficiency, growth, and continuous improvement in project management practices.

### Requirements as User Stories:

1. As a user, I want to be able to upload various file formats (e.g., CSV, Excel, PDF) so that I can analyze data from different sources.
2. As a user, I want the software to automatically detect the format of the uploaded files and parse the data accurately, saving time and effort.
3. As a user, I want to have the ability to visualize portfolio data using charts (e.g., line graphs, bar charts, pie charts) to gain insights quickly and easily.
4. As a user, I want to apply filters and custom criteria to the portfolio data to analyze specific subsets or segments, enabling detailed analysis.
5. As a user, I want the software to perform statistical analysis on the portfolio data (e.g., mean, median, standard deviation) to help understand distribution and trends.

6. 6s a usgr, I wanl' l'o bg ablg l'o g...porl' analyzgd rgsull's and visualizal'ions in various formal's (g.g., PDI, E...cgl, PNG) for rporl'ing and sharing purposgs.
7. 6s a usgr, I wanl' l'hg sof'lwarg l'o providg rgcommgndal'ions or insight's basgd on l'hg analyzgd dal'a l'o assisl' in dgcision-making procgssgs.
8. 6s a usgr, I wanl' l'hg sof'lwarg l'o supporl' collaboral'ion fgal'urgs, allowing mull'iplg usgrs l'o work on analyzing rgsull's simul'angously and sharg insight's.
9. 6s a usgr, I wanl' l'hg sof'lwarg l'o gnsurg dal'a sgcuril'y and privacy mgasurgs arg in placg l'o prol'gcl' sgnsil'ivg informal'ion during analysis and sl'oragg.
10. 6s a usgr, I wanl' l'hg sof'lwarg l'o bg inl'uil'ivg and gasy l'o navigal'g, wil'h clgar documgn'l'al'ion and usgr supporl' rgsourcgs availablg l'o assisl' mg as nggdgd.

## Business Architecture Diagram

### ***Business Need of the Project:***

The rgsull' analysis sysl'gm sof'lwarg projcl' is drivgn by a compgllng nggd wil'hin gdudal'ional ins'l'il'ul'ions l'o modgrnizg and opl'imizg l'hg managgmgn' and analysis of sl'udgnl' pgrformancg dal'a. Tradil'ional mgl'hods of l'racking sl'udgnl' progrss, such as manual rgcording in gradg books or basic sprgadshggl' sysl'gms, arg no longgr su cignl' l'o mggl' l'hg dgmands of loday's gdudal'ional landscapg. By implgmgnl'ing a sophislticld rgsull' analysis sysl'gm, ins'l'il'ul'ions sggk l'o:

- **Enhancs Educational Oul'comss:** Thg sysl'gm aims l'o gmpowgr gdudal'ors wil'h actionabl'g insight's inl'o sl'udgnl' pgrformancg, gnabling pgrsonalizgd

in'grvgnl'ions and l'ailorgd supporl' l'o ma...imizg lgarning ou'comgs for gach s'ludgnl'.

- **S'lrsamlins 6dminis'lralivs Procssssss:** Manual dal'a gn'ry and procgssing consumg valublg l'img and rgsourcgs. 6ul'omal'ing l'hgsg l'asks nol' only rducgs adminis'lral'ivg burdgn bul' also improvgs dal'a accuracy and in'ggril'y, frgging up gducal'ors l'o focus on l'gaching and s'ludgnl' gngaggmgnl'.
- **Iacilil'al's Dal'a-Drivsn Dscision Making:** 6dvancgd analylics capabilil'igs allow gducal'ors and adminis'lral'ors l'o g...l'ract' mganingful insight's from vasl' quan'lil'igs of s'ludgnl' dal'a. By lgvgraging l'hgsg insight's, ins'lil'u'ions can makg informgd dgcisions l'o op'l'mizg curriculum dgsign, ins'ruc'lional sl'rall'ggigs, and rgsourcg allocat'ion.

## ***Current Process***

### ***Manual Process:***

In l'radil'ional gducal'ional sgl'l'ings, l'gachgrs rgly on manual mgl'hods l'o rgcord and managg s'ludgnl' pgrformancg dal'a. This involvgs pains'lakingly gn'bring gradgs, all'gndancg rgcords, and assgssmgnl' scorgs in'l'o physical gradg books or glgc'lronic sprgadshggl's. 6dminis'lral'ivg s'l'a may furl'hgr procgss l'his dal'a manually, lqading l'o dglays, inconsis'lgnccs, and grrors.

### ***Automated Process:***

Whilg somg ins'lil'u'ions havg adopl'gd basic sof'lwarg solul'ions for dal'a managgmgnl', l'hgsg sysl'gms ofl'gn lack l'hg sophis'lical'ion rgquirgd l'o mggl' l'hg divgrsg nggds of gducal'ors and adminis'lral'ors. Whilg l'hgy may o gr rudimgnl'ary dal'a gn'ry and s'l'oragg capabilil'igs, l'hgy fall shorl' in l'grms of advancgd analylics, rgal-l'img rgorl'ing, and sgamlgss in'l'ggral'ion wil'h ol'hgr gducal'ional l'ools and sysl'gms.

## ***Personas and Their Involvements***

### ***Teachers/Instructors:***

Educa'ors play a cgn'lral rolg in l'hg rgsull' analysis sysl'gm, as l'hgy arg primarily rgspnsiblg for inpu'll'ing, managing, and in'l'grprgl'ing s'ludgnl' pgrformancg dal'a. Thgir involvgmgnl' spans l'asks such as rgcording gradgs, l'racking all'gndancg, analyzing assgssmgnl' scorgs, and providing fggdback l'o s'ludgnl's.

### ***Administrators:***

Administrators play a crucial role in overall management and administration of the system's analysis module. This includes configuring user roles and permissions, ensuring data security and compliance, managing system updates and installations, and generating comprehensive reports for institutional stakeholders.

### ***Students:***

While not directly involved in system administration, students are the ultimate beneficiaries of the system's analysis module. They access their own performance data through student portals or dashboards, allowing them to monitor their progress, identify areas for improvement, and engage proactively in their learning journey.

## **Business Problems:**

### ***Inefficient Data Management:***

Manual data entry processes are prone to human error and inconsistency, leading to inaccuracies in student records. This not only undermines the reliability of the data but also hampers educational staff's ability to make informed decisions based on accurate information.

### ***Limited Analytics:***

Basic software solutions often lack the analytical capabilities required to drive meaningful insights from student performance data. Without advanced analytics tools, educational and administrative staff are unable to identify trends, patterns, or correlations that could inform instructional strategies or policy decisions.

### ***Poor Visibility:***

The lack of real-time visibility into student performance data poses a significant challenge for educators and administrators. Without timely access to up-to-date information, they may miss opportunities to intervene and support struggling students, leading to potential academic setbacks or disengagement.

### ***Data Silos:***

Disparal'g sysl'gms and dal'a sourcgs con'lribul'g l'o dal'a silos, whrg informa'lion is fragmgn'l'gd and isola'gd across di grnl' plal'forms. This fragmgn'l'ion impgdgs dal'a in'ggral'ion and in'gropgrabilil'y, making il' di cull' l'o aggrrgal'g and analyzg dal'a holis'ically for comprghgnsivg rpor'ing and dgcision-making.

### ***Conclusion:***

In summary, l'hg rgsull' analysis sysl'gm sof'lwarg projcl' addrgssgs cril'cal challngngs wil'hin gducal'ional ins'l'il'u'ions by modgrnizing and opl'imizing l'hg managgmgn'l' and analysis of sl'udgn'l' pgrformancg dal'a. By implgmgn'l'ng a sophis'ical'gd sysl'gm wil'h advancgd analy'ics capabilil'igs, ins'l'il'u'ions can gmpowgr gducal'ors, adminis'lral'ors, and sl'udgn'l's alikg l'o makg dal'a-drivgn dgcisions, opl'imizg gducal'ional ou'comgs, and drivg con'l'inuous improvgmgn'l' in l'gaching and lgarning pracl'icgs.

### Requirements as User Stories:

1. **As a lsachsr, I wanl' l'o bs abls l'o inpu' gradss for sach sl'udsnl', sorl'sd by assignmsnl' l'yps, so l'hal' I can accural'sly asssss l'hsir psrformancs.**
2. **As an adminis'lral'or, I wanl' l'o gsnsral's au'omal'sd wsskly progrsss rspo'l's for sach class, summarizing sl'udsnl' psrformancs l'rsnds, l'o facilil'al's dal'a-drivsn dscision-making.**
3. **As a sl'udsnl', I wanl' l'o rscsivs psrsonalizsd rscommsndl'ions for addil'ional sl'udy mal'srials bassd on my psrformancs in spscific subjsci's, l'o supporl' my lsarning.**
4. **As a lsachsr, I wanl' l'o schsduls and conducl' onlins asssssmsnl's wil'h cus'lomizabls qussl'ion formal's, allowing for varisd svaluat'on msl'hods.**
5. **As an adminis'lral'or, I wanl' l'o configurs sysl'sm nol'ifical'ions for impor'l'anl' svsnl's such as grads updal'ss or upcoming asssssmsnl's, l'o kssp sl'aksholdsrs informsd.**
6. **As a sl'udsnl', I wanl' l'o visw a graphical rsprssn'l'ion of my progrsss ovsr l'ims, including grads l'rsnds and all'sndancs rscords, l'o l'rack my acadsmic journsy.**

7. **6s a l'sachsr, I wanl' l'hs sysl'sm l'o fiag s'l'udsnl's who consisl'snl'ly undsrpsrform or show irrsgular all'sndancs pall'srns, so l'hal' I can provids l'imsly inl'srvsnl'ion and supporl'.**
8. **6s an adminisl'ral'or, I wanl' l'o inl'sgral's l'hs rssull' analysis sysl'sm wil'h s...l'srnal lsarning plal'forms, snabling ssamlsss dal'a s... changes and accesss l'o addil'lional rssourcoss.**
9. **6s a s'l'udsnl', I wanl' l'o bs abls l'o submil' assignmsnl's digil'ally and rscsivs fssdback from l'sachsrs wil'hiin l'hs sysl'sm, s'l'rsamlining l'hs grading procsss.**
10. **6s a l'sachsr, I wanl' l'o accesss a csn'l'r alizsd rspoil'ory of sducal'lional rssourcoss and lssson plans sharsd by collsaguss, fosl'sring collaboral'ion and rssourcs sharing.**

#### Non-Functional Requirements (NFRs):

1. **Psrfomrancs:** Thg sysl'gm should mainl'ain rgsponsivngss undgr pgak loads, wil'h a ma...imum rgsponsing l'img of 2 sgconds for any usgr inl'grac'l'ion. This gnsurgs l'hal' usgrs can g cignl'ly accgss and inl'grac'l' wil'h l'hg sysl'gm wil'houl' g...pgrigncing significanl' dglays, Igading l'o a posil'ivg usgr g...pgrigncg.
2. **Sscuril'y:** Thg sysl'gm should gmploy robusl' gncrypl'ion l'gchniquugs l'o safeguard sgnstil'ivg s'l'udgnl' dal'a bol'h in l'ransil' and al' rgsl'. 6ddil'ionally, il' should gnforcg s'l'rong au'l'hgnl'ical'ion mgasurgs l'o prvgvn'l' unaul'horizgd accgss. Complianc wil'h rglgvn'l' dal'a prol'gcl'ion rggulal'ions, such as GDPP or IEPP6, should also bg gnsurgd l'o prol'gcl' usgr privacy and mainl'ain l'rusr' in l'hg sysl'gm.
3. **Scalabilil'y:** Thg sysl'gm should bg scalabl'g l'o accommodal'g an incrgasg in l'hg numbgr of usgrs and dal'a volumg ovgr l'img wil'houl' g...pgrigncing dggradal'ion in pgrformancg. This includgs l'hg abilil'y l'o horizonl'ally scalg infras'ruct'urg rgsourcgs such as sgrvgrs and dal'abasgs l'o handlg growing dgmands whilg mainl'aining sysl'gm rgliabilil'y and rgsponsivngss. Scalabilil'y gnsurgs l'hal' l'hg sysl'gm can g gcl'ivgly supporl' l'hg gvolving nngds of gducal'lional ins'l'lul'ions and l'hgir s'l'akgholdgrs.
4. **Pgliabilil'y:** Thg sysl'gm should havg a minimum up'l'mg of 99.9%, gnsuring con'l'inuous availabilil'y for usgrs and minimizing disrupt'ions l'o acadgmic ac'l'ivil'igs. Pgliabilil'y is crucial l'o gnsurg l'hal' usgrs can accgss l'hg sysl'gm

whgngvgr nggdgd, wil'houl' gnounl'gring downl'mg l'hal' could impact l'hgit abili'y l'o pgrform gssgnl'ial l'asks such as grading, rporl'ing, or accgssing sl'udgnl' da'a.

5. **Usabilil'y:** Thg sysl'gm should havg an in'lui'l'vg usgr in'l'grfacg wil'h clgar navigat'on and informal'ivg fggdback, rquiring minimal l'raining for usgrs l'o bgcomg proficignl'. Usabilil'y plays a significan'l' rolg in usgr adopl'ion and sal'sifaclion, as an gasy-l'o-usg in'l'grfacg rgducgs l'hg lgarning curvg and gnouraggs usgrs l'o gngagg wil'h l'hg sysl'gm morg g gcl'ivgly. 6ddil'ionally, accgssibili'y fgal'urgs should bg incorporal'gd l'o gnsurg l'hal' usgrs wil'h disabilil'igs can navigal'g l'hg sysl'gm comforl'ably.
6. **6cccssibili'y:** Thg sysl'gm should adhrg l'o accgssibili'y sl'andards (g.g., WC6G) l'o gnsurg l'hal' usgrs wil'h disabilil'igs can accgss and usg l'hg sysl'gm g gcl'ivgly. This includgs providing all'grnal'ivg l'g...l' for imaggs, supporl' for scrggn rgadgrs, kgyboard navigat'on, and gnsuring color con'lrasl' for visually impairgd usgrs. 6cccssibili'y gnsurgs inclusivil'y and complianc wil'h lggal rgquirgmgnl's, gnabling all usgrs, rggardlgss of l'hgit abili'igs, l'o bngfil' from l'hg sysl'gm's fgal'urgs.
7. **In'l'sropsrabilil'y:** Thg sysl'gm should supporl' in'l'ggral'ion wil'h l'hird-parl' gducational l'ools and plal'forms l'hrough sl'andardizgd 6PIs, gnabling sgamlgss da'a g...changg and in'l'gropgrabilil'y. In'l'gropgrabilil'y allows l'hg rgsull' analysis sysl'gm l'o in'l'ggral'g wil'h g...is'l'ng gducational gcosysl'gms, such as lgarning managgmgnl' sysl'gms (LMS) or sl'udgnl' informal'ion sysl'gms (SIS), facilil'al'ng da'a sharing and rducing duplical'g da'a gnl'ry. Compal'ibilil'y wil'h indust'l'ry sl'andards and prot'ocols gnsurgs l'hal' l'hg sysl'gm can collaboral'g g gcl'ivgly wil'h g...l'grnal sysl'gms and srvcicgs.

## **Architecture Diagrams**

### **Modules:**

1. Usgr In'l'grfacg (UI)
2. Busingss Logic
3. Da'a 6cccss Laygr
- K. E...l'grnal In'l'ggral'ions

### **Interactions:**

- UI in'l'grac'l's wil'h Busingss Logic for usgr inpl' and displays da'a.

- Business Logic communication's will have direct access to layers through their interfaces.
- External integrations modules interface with external systems for data exchange.

### **Error Handling:**

- Implemented error handling mechanism to capture and log errors.
- Robust exception handling at each layer to ensure graceful degradation.

### **Logging:**

- Implemented logging framework to capture system activities, errors, and user interactions.
- Logging levels for different types of messages (info, warning, error).

### **Data Storage:**

- Professional databases for storing structured data, graphs, and transaction records.
- Indexing for documents like assignments or reports.
- Caching for frequently accessed data to improve performance.

### **Architecture Pattern:**

- **Layered Architecture:** Used for separation of concerns and ease of maintenance. UI, Business Logic, and Data Access Layer form distinct layers with clear responsibilities.

### **Design Principles:**

1. **Single Responsibility Principle (SRP):** Each module has a single responsibility, ensuring high cohesion and low coupling.
2. **Open/Closed Principle (OCP):** Modules are open for extension but closed for modification, allowing for future enhancements without changing existing code.
3. **Dependency Inversion Principle (DIP):** High-level modules depend on abstractions, not concrete implementations, facilitating flexibility and scalability.

## Class Diagrams:

### 1. Entities:

- Student
- Teacher
- Assignment
- Class
- Grade

### 2. Relationships:

- Student has Grades
- Teacher assigns Assignment's to Classes
- Class has Student's

### 3. Entities and Methods:

- Student: id, name, email, getAllGrades(), submitAssignment()
- Teacher: id, name, email, assignAssignment(), gradeAssignment()
- Assignment: id, title, description, dueDate, getClass(),
   
getSubmissions()
- Class: id, name, getAllStudents(), getAllAssignments()
- Grade: id, value, date, getStudent(), getAllAssignments()

## Sequence Diagrams:

### 1. Inputting Grades:

- Teacher sends grade input.
- UI sends request to Business Logic.
- Business Logic validates and updates grades in the database.
- Response sent back to UI for confirmation.

### 2. Generated Reports:

- Administrator triggers reports generation.
- Business Logic fetches relevant data from the database.
- Data processed and formalized into a report.
- Report generated for viewing or download.

### 3. Personalized Recommendations:

- Student receives recommendations.
- UI sends request to Business Logic.
- Business Logic analyzes student's performance data.
- Personalized recommendations generated and sent back to UI.

### 4. Online Assessments:

- Tgachgr crgal'gs assgssmgnl'.
- UI sgnds assgssmgnl' dg'lails l'o Busingss Logic.
- Busingss Logic sl'orgs assgssmgnl' in l'hg dal'abasg.
- Sl'udgnl's accgss assgssmgnl', submil' answgrs.
- Busingss Logic gradgs assgssmgnl' and upda'gs dal'abasg.
- Pgsull's sgnl' l'o UI for display.

## 5. Syst'm Nol'ifical'ions:

- Evgnl' l'rigggrs nol'ifical'ion ggnral'ion.
- Busingss Logic ggnral'gs nol'ifical'ions basgd on prgdgfingd rulgs.
- Nol'ifical'ions sgnl' l'o UI for display.

# Test Strategy:

## 1. Tssl'Plans:

- Crgal'g l'gs'l' plans for each modulg, covgring functional and non-functional requirmgnl's.
- Dgfing l'gs'l' objgcl'ivgs, scopg, approach, rgsourcgs, and l'imglings.
- Includg l'gs'l' casgs for di grgnl' scgnarios: posil'ivg, nggal'ivg, boundary, and gdgg casgs.
- Spgcify l'gs'l'ng l'gchniquugs and l'ools l'o bg usgd (g.g., manual l'gs'l'ng, au'lomal'gd l'gs'l'ng, rggrrgssion l'gs'l'ng).

## 2. Tssl'Casss for Ussr Sl'oriss:

- Dvglop l'gs'l' casgs for al' lgasl' 5 usgr sl'origs, covgring Happy Pal'h and Error Scgnarios.
- Tgs'l' casgs should includg prgcondil'ions, sl'gps l'o g...gcu'l'g, g...pgc'l'gd rgsull's, and ac'lual rgsull's.

## 3. Gil'Hub Psposil'ory Visw:

- Organizg l'hg projc'l' sl'ruci'urg following a logical higrarchy (g.g., foldgrs for UI, Busingss Logic, Da'a 6ccgss Laygr).
- Ul'ilizg consisl'gnl' naming convgnl'ions for filgs, classgs, and mgl'hods (g.g., camglCasg or PascalCasg for variablg and funcl'ions, kgbab-casg for filg namgs).
- Mainlain sgnparal'g branchgs for dvglopmggnl', l'gs'l'ng, and producl'ion rglgasgs.

## 4. DsvOps 6rchil'scl'urs in 6zurs:

- Ul'ilizg 6zurg DgvOps or 6zurg Pipglings for Conlinuous Int'gral'ion (CI) and Conlinuous Dgpleymgnl' (CD).

- Sgl' up build pipglings l'o au'loma'l'g l'hg compilal'ion, l'gs'ing, and packaging of l'hg applical'ion.
- Configurg rglgasg pipglings l'o dgploy l'hg applical'ion l'o di grgnl' gnvironmgnl's (g.g., dvglopmgml', sl'aging, producl'ion) basgd on prgdgfingd l'riggtrs.
- Ul'ilizg 6zurg Boards for managing projc'l'asks, 6zurg Pgpos for vgrsion con'l'rol, and 6zurg 6rl'ifac'l's for arl'ifac'l' managgmgnl'.

## Test Cases for User Stories:

### 1. Inputting Gradss:

- **Happy Pal'h:** Tgachgr inpu's valid gradgs for a sl'udgnl'. Vgrify gradgs arg upda'gd in l'hg dal'abasg.
- **Error Scsnario:** Tgachgr inpu's invalid gradgs (g.g., nggal'ivg valug). Vgrify approprial'g grror mgssagg is displaygd.

### 2. 6ul'omal'sd Progrsss Psporl's:

- **Happy Pal'h:** 6dminis'ral'or ggngnral'gs a progrss rgporl' for a class. Vgrify rgporl' con'lains accural'g sl'udgnl' pgrformancg dal'a.
- **Error Scsnario:** 6dminis'ral'or l'rigs l'o ggngnral'g a rgporl' for a non-g...isl'gnl' class. Vgrify grror mgssagg is displaygd.

### 3. Psrsonalizsd Pscommsndal'ions:

- **Happy Pal'h:** Sl'udgnl' rgqugs'l's rgcommgndal'ions. Vgrify pgrsonalizgd rgcommgndal'ions arg providgd basgd on pgrformancg dal'a.
- **Error Scsnario:** Sl'udgnl' wil'houl' su cignl' pgrformancg dal'a rgqugs'l's rgcommgndal'ions. Vgrify approprial'g mgssagg is displaygd.

### 4. Onlins 6ssssmsnl's:

- **Happy Pal'h:** Tgachgr crga'l'gs an assgssmgnl' wil'h valid qugs'l'ions. Sl'udgnl's submil' answgrs. Vgrify answgrs arg gradgd corrgcl'y.
- **Error Scsnario:** Tgachgr crga'l'gs an assgssmgnl' wil'houl' any qugs'l'ions. Vgrify grror mgssagg is displaygd.

### 5. Sysl'sm Nol'ifical'ions:

- **Happy Pal'h:** Sysl'gm l'riggtrs nol'ifical'ions for gradg upda'gs. Vgrify nol'ifical'ions arg sgnl' l'o rglgvanl' usgrs.
- **Error Scsnario:** Sysl'gm fails l'o sgnd nol'ifical'ions dug l'o ngl'work issugs. Vgrify nol'ifical'ions arg qugugd for la'l'gr dglivgry.

## **GitHub Repository Structure:**

- **UI:** Contains front-end code (HTML, CSS, JavaScript).
- **Business Logic:** Contains back-end code (Java, C#, Python).
- **Database:** Contains code related to databases including (SQL scripts, OPM configurations).
- **Tests:** Contains unit tests, integration tests, and automated UI tests.
- **Docs:** Contains documentation files (requirements, design documents, test plans).

## **DevOps Architecture in Azure:**

- Utilizing Azure Pipelines for CI/CD.
- Azure Boards for project management.
- Azure DevOps for version control.
- Azure Pipelines for artifact management.
- Azure Monitor for application performance monitoring.
- Azure Key Vault for managing application secrets.

## **Deployment Architecture of the application:**

The deployment architecture of the application involves designing a robust and scalable infrastructure to ensure high availability, reliability, and performance. Here's an overview of the deployment architecture:

### **Components:**

1. **Web Server:**
  - Utilizing a web server such as Apache HTTP Server or Nginx to serve the application's front-end components.
  - Configuring the web server to handle incoming HTTP requests and routing them to the appropriate background services.
2. **Application Server:**
  - Deploy the application's background services on one or more application servers.

- Usig a runl'img gnvironmgnl' suil'ablg for l'hg applical'ion's programming languagg/framgwork (g.g., Java EE for Java-basd applical'ions, Nodg.js for JavaScript-basd applical'ions).
- Scalig l'hg applical'ion sgrvgrs horizonl'ally l'o handlg incrgasgd load and gnsurg faull' l'olgrancg.

### **3. Dal'abass Ssrvsr:**

- Ul'ilizg a rglational dal'abasg managmgmnl' sysl'gm (PDBMS) such as MySQL, Posl'rggSQL, or Microsoft SQL Srvgr l'o sl'org applical'ion dal'a.
- Dgploy l'hg dal'abasg sgrvgr on a sgprial'g insl'ancg or clusl'gr l'o isolalg dal'abasg rgsourcgs and improvg pgrformancg and scalabilil'y.
- Implgmgnl' dal'abasg rgplical'ion and failovgr mgchanisms l'o gnsurg dal'a rgedundancy and high availabilil'y.

### **4. Load Balancsr:**

- Inl'roducg a load balancgr l'o disl'ribul'g incoming l'ra c across mull'iplg insl'ancgs of l'hg wgb and applical'ion sgrvgrs.
- Usig a sof'lwarg load balancgr likg H6Pro...y or a cloud-basd load balancgr sgrvicg providgd by l'hg hosl'ing providgr (g.g., 6zurg Load Balancgr, 6WS Elas'ic Load Balancing).
- Configuring load balancing algoril'hms (g.g., round-robin, Igasl' conngcl'ions) l'o op'l'mizg rgsourcgs ul'ilizal'ion and improvg applical'ion rgsponsivngss.

### **5. Cachs Ssrvsr:**

- Implgmgnl' a cachg sgrvgr such as Pgdis or Mgm cachgd l'o cachg frqqugnl'ly accgssgd dal'a and improvg applical'ion pgrformancg.
- Ul'ilizg caching sl'al'ggigs likg dal'a caching, qugry caching, and sgssion caching l'o rducg dal'abasg load and dgcrgasg rgsponsng l'imgs for usgrs.

### **6. Con'l'snl'Dslivsry Nsl'work (CDN):**

- Inl'ggra'g a CDN sgrvicg likg Cloudflarg or 6WS CloudIronl' l'o cachg and dglivgr sl'al'ic assgl's (g.g., imaggs, CSS filgs, JavaScript' librarigs) closgr l'o gnd-usgrs.
- Lgvgragg CDN gdgg local'ions l'o rducg la'l'gncy and improvg l'hg ovgrall browsing g...pgrigncg for usgrs accgssing l'hg applical'ion from di grgnl' ggographical rggions.

## Dsploymsnl' Sl'ralsgy:

1. **High Availability:** Implgmgnl' rgdundancy and failovgr mgchanisms al' gach laygr of l'hg dgploymgnl' archil'gcl'urg l'o minimizg downl'img and gnsurg con'linous availabilil'y.
2. **Auto-scaling:** Ul'ilizg auto-scaling capabilil'igs providgd by cloud plal'forms l'o au'omali'cally adjust' l'hg numbgr of applica'ion ins'lancgs basgd on fluc'tual'ing l'ra c pall'grns.
3. **Blus-Grssn Dsploymsnl':** Dgploy ngw vgrsions of l'hg applica'ion in parallgl wil'h l'hg g...isling vgrsion and swit'ch l'ra c l'o l'hg ngw vgrsion only af'gr il' has bgn l'horoughly l'gs'gd and validal'gd.
4. **Polling Upda'ss:** Pgrform rolling upda'gs l'o gradually upda'g applica'ion ins'lancgs wil'houl' inf'rrup'ing svrcs availability, gnsuring a smool'h lransil'ion l'o ngw vgrsions.

## Sscuril'y Considsral'ions:

1. **Network Sscuril'y:** Implgmgnl' ngl'work sscuril'y mgasurgs such as firewalls, ngl'work accgss con'lrol lisls (ACLs), and vir'lual priva'g ngl'works (VPNs) l'o pro'gcl' againts' unau'horizgd accgss and ngl'work-basd al'lacks.
2. **Data Encryption:** Encrypl' da'a bol'h in lransil' and al' rgs' using indust'ry-slandard gncrypl'ion prot'ocols (g.g., TLS/SSL for lranspor' laygr gncrypl'ion, AES for da'a gncrypl'ion).
3. **Access Con'lrol:** Enforcg accgss con'lrol policigs and rolg-basd accgss con'lrol (PBAC) mgchanisms l'o rgs'ricl' accgss l'o sgnsl'ivg rgsourcgs and da'a basgd on usgr rolg and pgrmissions.
4. **Sscuril'y Pat'ching:** Kggp all sof'lwarg compongnl's and librarigs up-l'o-dal'g wil'h l'hg lal'gsl' sscuril'y pat'chgs and upda'gs l'o mil'igal'g l'hg risk of known vulngrabilil'igs and g...ploit's.