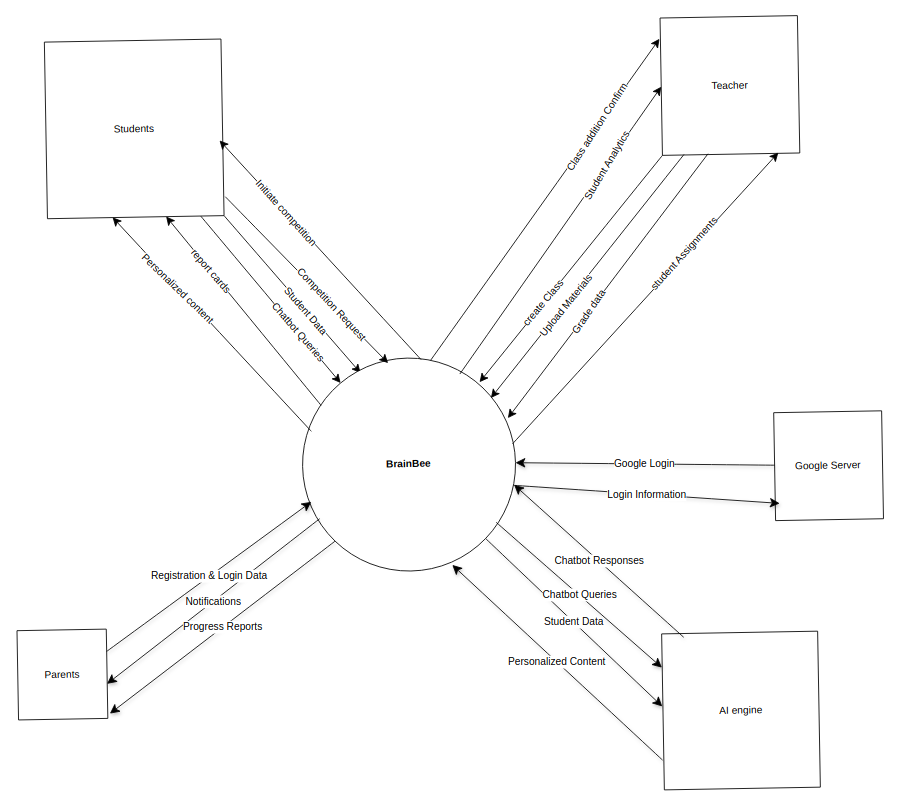
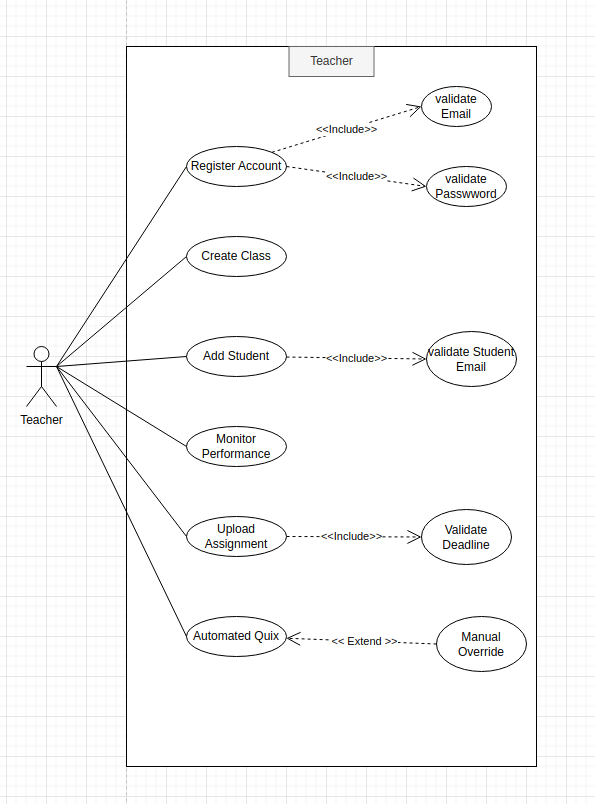
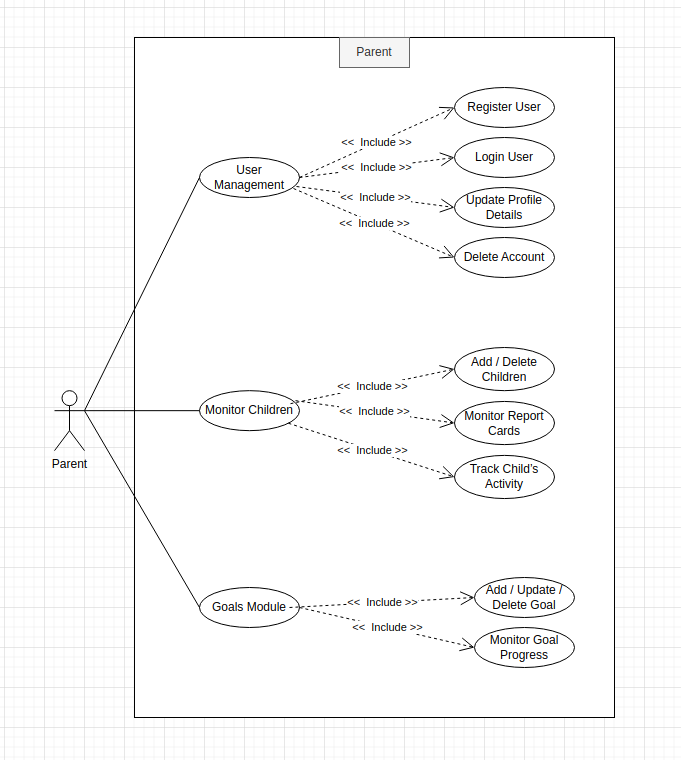
Event response table Template

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Event | Data Element | System State | Exception Condition | System Response | | | System State |
| In case of yes | In case of no | |
| FR-1 | User Registration |  |  |  |  | |  |  |

****

**Detailed Use Cases**

**UC-1: Teacher Registration**

|  |  |
| --- | --- |
| **Actors:** | Primary: Teacher  Secondary: Institutional Database, Email Server |
| **Trigger:** | The teacher selects the "Register" option on the BrainBee app. |
| **Preconditions:** | PRE-1: Teacher has the BrainBee app installed on their device.  PRE-2: Teacher has a stable internet connection.  PRE-3: Teacher has a valid email address |
| **Postconditions:** | POST-1: A new teacher account is created and stored in the system’s database.  POST-2: A confirmation message is displayed, and a verification email is sent. |
| **Normal Flow:** | 1.0 Register Account   1. The teacher opens the BrainBee app. 2. The teacher selects "Register" from the login screen. 3. Teacher enters:    a. Full name   b. Email address   c. Password (meeting complexity requirements)   d. Phone number (optional)   1. The teacher submits the registration form. 2. System validates:    a. Email uniqueness (BR-1)   b. Password strength (BR-2)   1. The system creates an account and sends a confirmation email. 2. Teacher receives a successful message. |
| **Alternative Flows:** | None |
| **Exceptions:** | E1: **Duplicate Email**   1. If the email is already registered, the system displays: "Email already exists. Log in or use another email." 2. Teacher returns to step 3b of Normal Flow.   E2: **Weak Password**   1. If the password does not meet complex rules, the system displays: "Password must include 8+ characters, uppercase, and a number." 2. Teacher returns to step 3c of Normal Flow. |
| **Business Rules:** | BR-1: Email addresses must be unique.  BR-2: Password must contain 8+ characters, 1 uppercase letter, and 1 number.  BR-3: Confirmation emails must be sent within 2 minutes of registration. |
| **Assumptions:** | 1. Teachers have access to a valid email inbox. 2. Teachers understand password complexity requirements. |

**UC-2: Create New Class**

|  |  |
| --- | --- |
| **Actors:** | Primary: Teacher  Secondary: Students |
| **Trigger:** | The teacher selects "Create New Class" from the Class Management section. |
| **Preconditions:** | PRE-1: Teacher is logged in.  PRE-2: Teacher has no duplicate class names (BR-1). |
| **Postconditions:** | POST-1: New class is created with a unique ID.  POST-2: Class appears in the teacher’s dashboard. |
| **Normal Flow:** | 1. Teacher navigates to "Class Management."  2. Teacher selects "Create New Class."  3. Teacher enters:   a. Class name (e.g., "Grade 10 Biology")   b. Subject   c. Grade level   d. Class schedule (optional)  4. Teacher clicks "Save."  5. System validates class name uniqueness (BR-1).  6. System generates a class code for student enrollment.  7. System displays: "Class created successfully." |
| **Alternative Flows:** | None |
| **Exceptions:** | E1: **Duplicate Class Name**   1. If the class name exists, the system displays: "Class name already in use. Please choose another." 2. Teacher returns to step 3a of Normal Flow. |
| **Business Rules:** | BR-1: Class names must be unique per teacher.  BR-2: Class codes must be auto-generated and alphanumeric (6 characters). |
| **Assumptions:** | 1. Teachers know their subject and grade-level details.  2. Classes are created before adding students. |

**UC-3: Add Students to a Class**

|  |  |
| --- | --- |
| **Actors:** | Primary: Teacher  Secondary: System, Student |
| **Trigger:** | Teacher selects "Add Students" in an existing class. |
| **Preconditions:** | PRE-1: Teacher has an active class. PRE-2: Student emails are valid (BR-1). |
| **Postconditions:** | POST-1: Students receive enrollment invitations. POST-2: Students appear in the class roster. |
| **Normal Flow:** | 1.0: Teacher creates class manually   1. Teacher navigates to "Class Management." 2. The teacher selects the target class. 3. Teacher clicks "Add Students." 4. Teacher manually enters student emails 5. The system validates email formats (BR-1). 6. System sends invitations to students. |
| **Alternative Flows:** | 2.0: Teacher creates class from csv file   1. Teacher navigates to "Class Management." 2. The teacher selects the target class. 3. Teacher clicks "Add Students." 4. Teacher uploads a CSV file with student data 5. The system validates email formats (BR-1). 6. System sends invitations to students. |
| **Exceptions:** | E1: **Duplicate Class Name**   1. If CSV files have errors, the system highlights invalid rows.   E2: **Invalid Email Format**   1. System detects invalid emails. 2. System displays: "Invalid email(s). Please correct and retry." |
| **Business Rules:** | BR-1: Student emails must follow standard email format (e.g., student@domain.com).  BR-2: Invitations expire after 7 days. |
| **Assumptions:** | 1. Teachers have access to student email lists.  2. Students have active email accounts. |

UC-4: Monitor Student Performance

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Teacher  Secondary Actor: System, Student |
| **Trigger:** | Teacher selects "View Performance" in the Assessment Module. |
| **Precondition:** | PRE-1: Students have completed quizzes/assignments. PRE-2: Performance data is available. |
| **Postcondition:** | POST-1: Teacher views analytics (e.g., scores, progress trends). POST-2: Teacher can adjust learning paths based on data. |
| **Normal Flow** | 1. Teacher navigates to "Assessment Module." 2. Teacher selects a class and student. 3. System displays:  a. Quiz/assignment scores  b. Progress graphs  c. AI-generated insights (e.g., "Student struggles with Algebra") 4. Teacher selects "Create Learning Path" to assign remedial materials. |
| **Alternative Flow** | None |
| **Exceptions** | E1: **Data Fetch Failure**   1. System fails to load data. 2. System displays: "Error loading data. Try again later." |
| **Business Rules** | BR-1: Data updates in real-time after quiz submissions.  BR-2: AI insights are generated only if ≥3 assessments are completed |
| **Assumptions:** | 1. Quizzes/assignments are graded automatically by the system. 2. Teachers use data to personalize instruction. |

Use Case 5: Upload Assignment

**Use Case ID:** UC-5

|  |  |
| --- | --- |
| **Actors:** | **Primary Actor: Teacher Secondary Actor: System** |
| **Trigger:** | Teacher selects "Upload Assignment" in a class. |
| **Precondition:** | PRE-1: Teacher is logged in. PRE-2: Class exists. |
| **Postcondition:** | POST-1: Assignment is visible to students. POST-2: Deadline reminders are scheduled |
| **Normal Flow** | 1. Teacher navigates to "Class Management." 2. Teacher selects a class. 3. Teacher clicks "Upload Assignment." 4. Teacher enters:  a. Assignment title  b. Instructions (text/PDF)  c. Deadline  d. Attachments (optional) 5. Teacher clicks "Publish." 6. System validates deadline (must be future date). 7. System sends notifications to students. |
| **Alternative Flow** | None |
| **Exceptions** | **E1: File Upload Failure** 1. System rejects large files (>10MB). 2. System displays: "File size exceeds limit. Max 10MB." |
| **Business Rules** | BR-1: Assignments must include a title and deadline. BR-2: Notifications are sent 24 hours before the deadline. |
| **Assumptions:** | 1. Teachers prepare assignments in compatible formats (PDF, DOCX). |

Use Case 6: Automated Quiz Grading with AI Feedback

**Use Case ID:** UC-6

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Teacher Secondary Actor: System, AI Grading Model |
| **Trigger:** | The teacher uploads a quiz for grading. |
| **Precondition:** | PRE-1: Quiz is in a supported format (PDF, DOCX, or app-generated). PRE-2: Answer key is provided, or AI model is pre-trained on the subject. |
| **Postcondition:** | POST-1: All student submissions are graded. POST-2: AI-generated feedback is attached to each quiz. |
| **Normal Flow** | 1. Teacher uploads a quiz file or creates one in-app. 2. Teacher selects "Auto-Grade with AI." 3. System processes submissions using:  a. **NLP** for essay/short-answer grading.  b. **Rule-based scoring** for multiple-choice. 4. AI highlights common errors (e.g., "5 students missed Question 3"). 5. The system generates per-student feedback (e.g., "Revise topic: Quadratic Equations"). 6. Grades and feedback are published to student dashboards. |
| **Alternative Flow** | 6.1 Manual Override 1. Teacher reviews AI-graded quizzes. 2. Teacher edits scores/feedback and clicks "Re-publish." |
| **Exceptions** | E1: Unsupported Question Type 1. AI flags ungradable questions (e.g., diagrams). 2. System alerts teacher: "Manually grade 2/10 questions." |
| **Business Rules** | BR-1: AI grading accuracy must exceed 90% for multiple-choice. BR-2: Feedback must align with curriculum standards. |
| **Assumptions:** | 1. Quizzes are structured clearly (e.g., numbered questions).  2. Teachers review AI feedback before publishing. |

**Use Case 7: Student Self-Registration**

|  |  |
| --- | --- |
| **Actors:** | Primary Actor: Student Secondary Actor: Institutional Database, Email Server |
| **Trigger:** | Students select the "Join Class" option on the BrainBee app.. |
| **Precondition:** | PRE-1: Student has the BrainBee app installed PRE-2: Student possesses a valid class code from the teacher. |
| **Postcondition:** | POST-1: Student account is created and linked to the class.  POST-2: Confirmation email is sent to the student. |
| **Normal Flow** | 1. Student opens the app and selects "Join Class."  2. Student enters class code, email, and password.  3. System validates class code and email uniqueness.  4. Account is created, and student is enrolled in the class. |
| **Alternative Flow** | 1. If the code is invalid, the system displays: "Invalid code. Try again."  Students return to step 2. |
| **Exceptions** | The system detects an expired code and notifies the student. |
| **Business Rules** | BR-1: Class codes expire after 14 days.  BR-2: Students must use institutional email domains. |
| **Assumptions:** | Teachers share class codes with students.  Students have access to email. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTIONAL REQUIREMENTS FOR TEACHER\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |
| --- | --- |
| **Identifier** | FR-T1 |
| **Title** | Account.Register |
| **Requirement** | The teacher shall be able to register an account in the BrainBee app by providing a full name, email address, password (meeting complex requirements), and an optional phone number. |
| **Source** | Sohail |
| **Rationale** | Teachers need an account to access classroom management tools and analytics, requiring a secure registration process. |
| **Business Rule** | BR-1: Email addresses must be unique.  BR-2: Password must contain 8+ characters, 1 uppercase letter, and 1 number. |
| **Dependencies** | FR-T1-2, FR-T1-3 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T1-2 |
| **Title** | Account.Validate.Email |
| **Requirement** | Upon entering an email address, the system shall validate that the email is not already registered in the institutional database. |
| **Source** | Sohail |
| **Rationale** | Ensures uniqueness of teacher accounts to prevent duplication and maintain data integrity. |
| **Business Rule** | BR-1: Email addresses must be unique. |
| **Dependencies** | FR-T1-4 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T1-3 |
| **Title** | Account.Validate.Password |
| **Requirement** | Upon entering a password, the system shall validate that it contains at least 8 characters, 1 uppercase letter, and 1 number. |
| **Source** | Sohail |
| **Rationale** | Enforces password complexity to enhance account security. |
| **Business Rule** | BR-2: Password must contain 8+ characters, 1 uppercase letter, and 1 number. |
| **Dependencies** | FR-T1-5 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | **FR-T1-4** |
| **Title** | Account.DuplicateEmail.Error |
| **Requirement** | If the email is already registered, the system shall display an error message: "Email already exists. Log in or use another email." |
| **Source** | Sohail |
| **Rationale** | Informs the teacher of email duplication and provides clear next steps. |
| **Business Rule** | BR-1: Email addresses must be unique. |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | **FR-T1-5** |
| **Title** | Account.WeakPassword.Error |
| **Requirement** | If the password does not meet complexity requirements, the system shall display an error message: "Password must include 8+ characters, uppercase, and a number." |
| **Source** | Sohail |
| **Rationale** | Guides the teacher to create a secure password meeting system standards. |
| **Business Rule** | BR-2: Password must contain 8+ characters, 1 uppercase letter, and 1 number. |
| **Dependencies** | none |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | **FR-T1-6** |
| **Title** | Account.Confirmation |
| **Requirement** | Upon successful registration, the system shall store the teacher’s account in the database, display a confirmation message, and send a verification email within 2 minutes. |
| **Source** | Sohail |
| **Rationale** | Verifies successful account creation and ensures the teacher can confirm their email. |
| **Business Rule** | BR-3: Confirmation emails must be sent within 2 minutes of registration. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| Identifier | **FR-T1-7** |
| Title | Account.NoConnection.Error |
| Requirement | If internet connectivity is lost during registration, the system shall display an error message: "Connection lost. Please check your internet and try again." |
| Source | Sohail |
| Rationale | Notifies the teacher of connectivity issues and prompts corrective action. |
| Business Rule | None |
| Dependencies | None |
| Priority | Medium |

UC-2: Create New Class

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-1** |
| **Title** | Class.Create |
| **Requirement** | The teacher shall be able to create a new class by providing a class name, subject, grade level, and an optional class schedule. |
| **Source** | Sohail |
| **Rationale** | Enables teachers to establish virtual classrooms for instructional purposes. |
| **Business Rule** | BR-1: Class names must be unique per teacher. |
| **Dependencies** | FR-T2-2 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-2** |
| **Title** | Class.Validate.Name |
| **Requirement** | Upon entering a class name, the system shall validate that the name is unique for the teacher’s account. |
| **Source** | Sohail |
| **Rationale** | Prevents confusion by ensuring no duplicate class names exist for a teacher. |
| **Business Rule** | BR-1: Class names must be unique per teacher. |
| **Dependencies** | FR-T2-3 |
| **Priority** | High |

l

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-3** |
| **Title** | Class.DuplicateName.Error |
| **Requirement** | If the class name already exists, the system shall display an error message: "Class name already in use. Please choose another." |
| **Source** | Sohail |
| **Rationale** | Informing the teacher of name conflicts and prompts a new selection. |
| **Business Rule** | BR-1: Class names must be unique per teacher. |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-4** |
| **Title** | Class.DuplicateName.ErrorClass.Generate.Code |
| **Requirement** | Upon successful class creation, the system shall generate a unique 6-character alphanumeric class code for student enrollment. |
| **Source** | Sohail |
| **Rationale** | Provides a secure method for students to join the class. |
| **Business Rule** | BR-2: Class codes must be auto-generated and alphanumeric (6 characters). |
| **Dependencies** | FR-T2-5 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-5** |
| **Title** | Class.Confirmation |
| **Requirement** | Upon successful class creation, the system shall display a message: "Class created successfully" and add the class to the teacher’s dashboard. |
| **Source** | Sohail |
| **Rationale** | Confirms class creation and ensures visibility on the dashboard. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | **FR-T2-6** |
| **Title** | Class.SaveFailure.Error |
| **Requirement** | If the system fails to save the class to the database, it shall display an error message: "Error saving class. Try again." |
| **Source** | Sohail |
| **Rationale** | Alerts the teacher to retry the operation due to a database issue. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

UC-3: Add Students to a Class

This section defines requirements for enrolling students into classes.

|  |  |
| --- | --- |
| **Identifier** | FR-T3-1 |
| **Title** | Students.Add.Manual |
| **Requirement** | The teacher shall be able to manually enter student email addresses to add them to an existing class. |
| **Source** | Sohail |
| **Rationale** | Allows teachers to individually enroll students using their emails. |
| **Business Rule** | BR-1: Student emails must follow standard email format (e.g., student@domain.com). |
| **Dependencies** | FR-T3-3 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-2 |
| **Title** | Students.Add.Bulk |
| **Requirement** | The teacher shall be able to upload a CSV file containing student data to add multiple students to an existing class. |
| **Source** | Sohail |
| **Rationale** | Facilitates efficient enrollment of multiple students at once. |
| **Business Rule** | BR-1: Student emails must follow standard email format (e.g., student@domain.com). |
| **Dependencies** | FR-T3-3 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-3 |
| **Title** | Students.Validate.Email |
| **Requirement** | The system shall validate that each student email follows the standard format (e.g., student@domain.com) before sending invitations. |
| **Source** | Sohail |
| **Rationale** | Ensures only valid emails receive enrollment invitations. |
| **Business Rule** | BR-1: Student emails must follow standard email format. |
| **Dependencies** | FR-T3-5 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-4 |
| **Title** | Students.BulkUpload.Error |
| **Requirement** | If the uploaded CSV file contains errors, the system shall highlight invalid rows and allow the teacher to correct and re-upload the file. |
| **Source** | Sohail |
| **Rationale** | Enables the teacher to fix errors in bulk uploads without restarting the process. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-5 |
| **Title** | Students.InvalidEmail.Error |
| **Requirement** | If an email format is invalid, the system shall display an error message: "Invalid email(s). Please correct and retry." |
| **Source** | Sohail |
| **Rationale** | Notifies the teacher of invalid emails and prompts correction. |
| **Business Rule** | BR-1: Student emails must follow standard email format. |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-6 |
| **Title** | Students.SendInvitations |
| **Requirement** | Upon successful validation, the system shall send enrollment invitations to student emails that expire after 7 days and update the class roster. |
| **Source** | Sohail |
| **Rationale** | Ensures students receive invitations and are added to the class roster. |
| **Business Rule** | BR-2: Invitations expire after 7 days. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T3-7 |
| **Title** | Students.Duplicate.Skip |
| **Requirement** | If a student is already enrolled, the system shall skip their email and display a message: "X students added successfully. Y duplicates skipped." |
| **Source** | Sohail |
| **Rationale** | Prevents redundant enrollments and informs the teacher of the outcome. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

Add Students to a Class

4.4 UC-4: Monitor Student Performance

|  |  |
| --- | --- |
| **Identifier** | FR-T4-1 |
| **Title** | Performance.View |
| **Requirement** | The teacher shall be able to view a real-time analytics dashboard displaying quiz/assignment scores, progress graphs, and AI-generated insights for a selected student in a class. |
| **Source** | Sohail |
| **Rationale** | Provides teachers with data to track and adjust student learning paths. |
| **Business Rule** | BR-1: Data updates in real-time after quiz submissions. BR-2: AI insights are generated only if ≥3 assessments are completed. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T4-2 |
| **Title** | Performance.NoData.Error |
| **Requirement** | If no performance data is available, the system shall display a message: "No performance data found. Assign quizzes first." |
| **Source** | Sohail |
| **Rationale** | Informs the teacher that assessments are required to generate analytics. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T4-3 |
| **Title** | Performance.DataFailure.Error |
| **Requirement** | If the system fails to load performance data, it shall display an error message: "Error loading data. Try again later." |
| **Source** | Sohail |
| **Rationale** | Alerts the teacher to retry due to a data retrieval issue. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T4-4 |
| **Title** | Performance.LearningPath |
| **Requirement** | The teacher shall be able to assign remedial materials by selecting "Create Learning Path" based on performance analytics. |
| **Source** | Sohail |
| **Rationale** | Enables personalized instruction based on student performance insights. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |

UC-5: Upload Assignment

|  |  |
| --- | --- |
| **Identifier** | FR-T5-1 |
| **Title** | Assignment.Upload |
| **Requirement** | The teacher shall be able to upload an assignment by providing a title, instructions (text or PDF), deadline, and optional attachments. |
| **Source** | Sohail |
| **Rationale** | Allow teachers to distribute assignments to students efficiently. |
| **Business Rule** | BR-1: Assignments must include a title and deadline. |
| **Dependencies** | FR-T5-2 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T5-2 |
| **Title** | Assignment.Validate.Deadline |
| **Requirement** | The system shall validate that the assignment deadline is a future date before publishing. |
| **Source** | Sohail |
| **Rationale** | Ensure assignments are relevant and actionable for students. |
| **Business Rule** | BR-1: Assignments must include a title and deadline. |
| **Dependencies** | FR-T5-4 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T5-3 |
| **Title** | Assignment.Edit |
| **Requirement** | The teacher should be able to edit assignment details before the deadline and republish the updated assignment. |
| **Source** | Sohail |
| **Rationale** | Allows flexibility to correct or refine assignments prior to the deadline. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T5-4 |
| **Title** | Assignment.PastDeadline.Error |
| **Requirement** | If the deadline is in the past, the system shall display an error message: "Deadline must be a future date." |
| **Source** | Sohail |
| **Rationale** | Prevents invalid deadlines from being set. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T5-5 |
| **Title** | Assignment.Publish |
| **Requirement** | Upon successful validation, the system shall publish the assignment, make it visible to students, and schedule a notification 24 hours before the deadline. |
| **Source** | Sohail |
| **Rationale** | Ensure students receive assignments and timely reminders. |
| **Business Rule** | BR-2: Notifications are sent 24 hours before the deadline. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T5-6 |
| **Title** | Assignment.FileSize.Error |
| **Requirement** | If an uploaded file exceeds 10MB, the system shall display an error message: "File size exceeds limit. Max 10MB." |
| **Source** | Sohail |
| **Rationale** | Enforces file size limits to maintain system performance. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

UC-6: Automated Quiz Grading with AI Feedback

|  |  |
| --- | --- |
| **Identifier** | FR-T6-1 |
| **Title** | Quiz.Upload |
| **Requirement** | The teacher shall be able to upload a quiz in PDF, DOCX, or app-generated format, including an answer key or relying on a pre-trained AI model. |
| **Source** | Sohail |
| **Rationale** | Enables automated grading by providing quiz content to the system. |
| **Business Rule** | None |
| **Dependencies** | FR-T6-2 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T6-2 |
| **Title** | Quiz.AutoGrade |
| **Requirement** | The system shall process student submissions using NLP for essay/short-answer questions and rule-based scoring for multiple-choice, achieving >95% accuracy for multiple-choice grading. |
| **Source** | Sohail |
| **Rationale** | Automates grading to save teacher time while maintaining accuracy. |
| **Business Rule** | BR-1: AI grading accuracy must exceed 95% for multiple-choice. |
| **Dependencies** | FR-T6-3 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T6-3 |
| **Title** | Quiz.Feedback.Generate |
| **Requirement** | The system shall generate AI-based feedback for each student (e.g., "Revise topic: Quadratic Equations") and highlight common errors (e.g., "5 students missed Question 3"). |
| **Source** | Sohail |
| **Rationale** | Provides actionable insights for students and teachers. |
| **Business Rule** | BR-2: Feedback must align with curriculum standards. |
| **Dependencies** | FR-T6-4 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T6-4 |
| **Title** | Quiz.Publish.Grades |
| **Requirement** | The system shall publish grades and feedback to student dashboards upon completion of grading. |
| **Source** | Sohail |
| **Rationale** | Ensure students receive timely results and feedback. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T6-5 |
| **Title** | Quiz.ManualOverride |
| **Requirement** | The teacher shall be able to review AI-graded quizzes, edit scores or feedback, and republish the updated results. |
| **Source** | Sohail |
| **Rationale** | Allows teachers to correct AI errors or adjust grades as needed. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T6-6 |
| **Title** | Quiz.UnsupportedQuestion.Error |
| **Requirement** | If the AI cannot grade certain questions (e.g., diagrams), the system shall alert the teacher: "Manually grade 2/10 questions." |
| **Source** | Sohail |
| **Rationale** | Ensure teachers address ungradable questions manually. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

UC-7: Student Self-Registration

|  |  |
| --- | --- |
| **Identifier** | FR-T7-1 |
| **Title** | Student.Register |
| **Requirement** | The student shall be able to register an account and join a class by entering a class code, email, and password in the BrainBee app. |
| **Source** | Sohail |
| **Rationale** | Enables students to access classes independently using a teacher-provided code. |
| **Business Rule** | BR-2: Students must use institutional email domains. |
| **Dependencies** | FR-T7-2 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T7-2 |
| **Title** | Student.Validate.Code |
| **Requirement** | The system shall validate that the entered class code matches an active class and is not expired (valid for 14 days). |
| **Source** | Sohail |
| **Rationale** | Ensures only valid, current codes allow enrollment. |
| **Business Rule** | BR-1: Class codes expire after 14 days. |
| **Dependencies** | FR-T7-4 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T7-3 |
| **Title** | Student.Validate.Email |
| **Requirement** | The system shall validate that the student’s email is unique and uses an institutional domain. |
| **Source** | Sohail |
| **Rationale** | Maintains account uniqueness and ensures institutional affiliation. |
| **Business Rule** | BR-2: Students must use institutional email domains. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-T7-4 |
| **Title** | Student.InvalidCode.Error |
| **Requirement** | If the class code is invalid or expired, the system shall display an error message: "Invalid code. Try again." |
| **Source** | Sohail |
| **Rationale** | Notifies the student of an incorrect or expired code and prompts re-entry. |
| **Business Rule** | BR-1: Class codes expire after 14 days. |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-T7-5 |
| **Title** | Student.Confirmation |
| **Requirement** | Upon successful registration, the system shall link the student to the class, store the account, and send a confirmation email. |
| **Source** | Sohail |
| **Rationale** | Confirms enrollment and notifies the student via email. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Parent APP  
**3.1.2 Detailed Use Cases**

### **UC-1: Register Account** A user (parent) registers for an account to access the Parent App functionalities. They must provide essential details such as name, username, email ID, phone number, and a password. The system ensures the uniqueness of the username and email before saving the account.

|  |  |
| --- | --- |
| **Actors:** | Primary: Parent  Secondary: System, Email Server |
| **Trigger:** | Parent selects the "Register" option on the BrainBee app. |
| **Precondition:** | PRE-1: Parent has the BrainBee app installed.  PRE-2: Parent has a valid email address.  PRE-3: Parent has a stable internet connection. |
| **Postcondition:** | POST-1: A new parent account is created and stored in the system’s database.  POST-2: A verification email is sent. |
| **Normal Flow** | 1. parents open the BrainBee app. 2. Parent selects "Register." 3. Parent enters: a. Full name b. Email address c. Password d. Phone number (optional) 4. Parent submits the registration form. 5. System validates: a. Email uniqueness (BR-1) b. Password strength (BR-2) 6. The system creates an account and sends a confirmation email. 7. Parents receive a success message. |
| **Alternative Flow** | 1.1 Duplicate Email   1. If the email is already registered, the system displays: "Email already exists. Log in or use another email." 2. Parent returns to step 3b of the Normal Flow.   1.2 Weak Password   1. If the password does not meet complex rules, the system displays: "Password must include 8+ characters, uppercase, and a number." 2. Parent returns to step 3c of the Normal Flow. |
| **Exceptions** | E1: Internet Connection Lost   1. The system detects loss of connectivity during registration. 2. System displays: "Connection lost. Please check your internet and try again." 3. Use case terminates. |
| **Business Rules** | BR-1: Email addresses must be unique.  BR-2: Password must contain 8+ characters, 1 uppercase letter, and 1 number.  BR-3: Confirmation emails must be sent within 2 minutes of registration. |
| **Assumptions:** | 1. Parents have access to a valid email inbox. 2. Parents understand password complexity requirements. |

### **Use Case 2: Monitor Child's Progress**

**Use Case ID:** UC-P2

#### **Description**

Allows parents to monitor their child's academic performance, quiz results, leaderboard ranking, and activity history.

|  |  |
| --- | --- |
| **Actors:** | Primary: Parent  Secondary: System, Child Profile |
| **Trigger:** | Parent selects "View Child Progress" from the dashboard. |
| **Precondition:** | PRE-1: Parent is logged in.   * PRE-2: Child profile is linked to the parent account. |
| **Postcondition:** | POST-1: Parent can view child’s academic performance.   * POST-2: Weekly and monthly reports are generated. |
| **Normal Flow** | Parent navigates to the "Monitor Child" section.   1. Parent selects a specific child profile. 2. System displays: a. Report cards b. Quiz scores c. Leaderboard ranking d. Activity history 3. Parent can filter reports by date or subject. |
| **Alternative Flow** | 2.1 No Data Available   * If the child has no quiz history, the system displays: "No data available. Encourage your child to take quizzes." |
| **Exceptions** | E1: System Fails to Load Data   * System detects an error fetching child data. * System displays: "Error retrieving data. Try again later." |
| **Business Rules** | BR-1: Progress reports must be updated in real-time. |
| **Assumptions:** | The child has participated in quizzes and assessments. |

### **Use Case 3: Set Learning Goals for Child**

**Use Case ID:** UC-P3

#### **Description**

Allows parents to set academic goals for their children and monitor progress towards completion.

|  |  |
| --- | --- |
| **Actors:** | Primary: Parent  Secondary: System, Child |
| **Trigger:** | Parent selects "Set Goals" from the parent dashboard. |
| **Precondition:** | PRE-1: Parent is logged in.  PRE-2: Child profile is linked to the parent account. |
| **Postcondition:** | POST-1: A goal is created for the child.  POST-2: System tracks goal completion progress. |
| **Normal Flow** | 1. Parent navigates to the "Goals" module. 2. Parents select "Create New Goal." 3. Parent enters: a. Goal name (e.g., "Complete 5 quizzes this week") b. Target completion date c. Reward (optional, e.g., virtual coins) 4. Parent saves the goal. 5. System tracks progress and updates completion status. |
| **Alternative Flow** | None |
| **Exceptions** | E1: System Fails to Save Goal   1. System displays: "Error saving goal. Try again." |
| **Business Rules** | BR-1: Goals must have a target completion date |
| **Assumptions:** | Parents actively monitor and update goals. |

**Functional Requirements for Paren**

|  |  |
| --- | --- |
| **Identifier** | FR-P1 |
| **Title** | Account.Register |
| **Requirement** | The system allow a parent to register an account in the BrainBee app by providing their full name, email address, password, and an optional phone number. |
| **Source** | Sohail |
| **Rationale** | Parents need an account to monitor their child's academic performance. |
| **Business Rule** | BR-1, BR-2, BR-3 |
| **Dependencies** | FR-P1-2, FR-P1-3 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P1-2 |
| **Title** | Account.Validate.Email |
| **Requirement** | The system shall validate that the email entered by the parent is unique before registration. |
| **Source** | Sohail |
| **Rationale** | Ensures uniqueness of accounts to prevent duplication. |
| **Business Rule** | BR-1 |
| **Dependencies** | FR-P1-4 |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P1-3 |
| **Title** | Account.Validate.Password |
| **Requirement** | The system shall ensure that the password meets complex requirements (at least 8 characters, 1 uppercase letter, and 1 number). |
| **Source** | Sohail |
| **Rationale** | Enforces secure password practices. |
| **Business Rule** | BR-2 |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P1-4 |
| **Title** | Account.DuplicateEmail.Error |
| **Requirement** | If a parent attempts to register with an already registered email, the system shall display an error message: "Email already exists. Log in or use another email." |
| **Source** | Sohail |
| **Rationale** | It prevents duplicate accounts. |
| **Business Rule** | BR-1 |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-P1-5 |
| **Title** | Account.WeakPassword.Error |
| **Requirement** | If the password does not meet complexity rules, the system shall display: "Password must include 8+ characters, uppercase, and a number." |
| **Source** | Sohail |
| **Rationale** | Ensures account security. |
| **Business Rule** | BR-2 |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-P2 |
| **Title** | Monitor.ChildProgress |
| **Requirement** | The system allow parents to view their child's academic progress, including quiz scores, leaderboard position, and activity history. |
| **Source** | Sohail |
| **Rationale** | Enables parents to track their child's learning journey. |
| **Business Rule** | BR-1: Progress reports must be updated in real-time. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P2-1 |
| **Title** | Child.ViewReportCard |
| **Requirement** | The system shall allow parents to view the child’s report card for specific subjects. |
| **Source** | Sohail |
| **Rationale** | Provides parents with detailed academic insights. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P2-2 |
| **Title** | Child.ViewLeaderboard |
| **Requirement** | The system shall allow parents to view their child's leaderboard ranking. |
| **Source** | Sohail |
| **Rationale** | Helps parents understand their child’s competitive standing. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-P2-3 |
| **Title** | Child.WeeklyReport |
| **Requirement** | The system shall generate and send weekly and monthly reports to parents. |
| **Source** | Sohail |
| **Rationale** | Keeps parents informed about long-term progress. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-P3 |
| **Title** | Parent.SetGoals |
| **Requirement** | The system shall allow parents to set academic goals for their children and track completion status. |
| **Source** | Sohail |
| **Rationale** | Encourages goal-based learning. |
| **Business Rule** | BR-1: Goals must have a target completion date. |
| **Dependencies** | None |
| **Priority** | High |

|  |  |
| --- | --- |
| **Identifier** | FR-P3-1 |
| **Title** | Parent.AssignRewards |
| **Requirement** | The system shall allow parents to assign rewards (e.g., virtual coins) for goal achievements. |
| **Source** | Sohail |
| **Rationale** | Motivates children to complete learning objectives. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | Medium |

|  |  |
| --- | --- |
| **Identifier** | FR-P3-2 |
| **Title** | Parent.TrackGoalCompletion |
| **Requirement** | The system shall allow parents to monitor progress towards assigned goals. |
| **Source** | Sohail |
| **Rationale** | Ensures goal achievement tracking. |
| **Business Rule** | None |
| **Dependencies** | None |
| **Priority** | High |