Please first complete participant information form (Table 1), then follow the instruction and finally fill the user experience form (Table 2).

Table 1. Participant information form

Participant Information
1. What is your gender?
☐ Woman ☐ Man ☐ I prefer not to say
2. What is your last degree? ----------
3. How familiar are you with model-driven software engineering?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
4. How much experience do you have in the modeling?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
5. How many years do you have experience in software modeling? ----------
6. How familiar are you with UML Class diagram?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
Background Information (in modeling tools)
7. How much modeling experience have you had with graphical modeling environments?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
8. How much modeling experience have you had with tree modeling editors?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
9. How much modeling experience have you had in text modeling editors?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
10. How much modeling experience have you had in desktop graphical editors such as Papyrus?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much
11. How much modeling experience have you had in graphic editors on the web?
☐ Very low ☐ Low ☐ Average ☐ Much ☐ Very much

University management system specification

The system proposed aims at implementing features to help professors, student, and the university in itself to ease their workload in terms of managing various aspects of the university. The system aims at providing tools to professors that will streamline the calculation of grades and grading curve keeping in mind the preferences of the professor on how to grade. This system aims to develop grading algorithm to ensure fair grading. Tools for professor will also provide the functionality to communicate with students for announcements, sharing coursework materials etc.
The system will include tools for students to help them in registration for courses they wish to take. The student will be able to sign up on wait list if the class capacity is currently full. The wait list will act as a queue for students wishing to attend the course. The students will be notified by email as and when the seat is up for grabs. Apart from tools for professors and students, the system will incorporate functionalities for departments in the university offering different courses to add, remove, modify course information. Course information will include parameters such as teaching professor, credits, duration etc. The system will also include a top-level admin functionality which will be able to manage all the departments, professors and students. The list of functionalities in not exhaustive and some functionalities might be removed or added depending upon the feasibility and necessity.

Functional requirements

1. Use Cases for User Authorization

1.1: Login

Use case name: Login
Actor: System User
Flow of events:
1. The user starts the application.
2. The system presents the login window to the user.
3. The user enters its username and password details in the provided fields.
4. The system tries to authenticate the user.
5. If the user is authenticated, an appropriate interface is shown to the user.

Alternate flow of events 1:
1. At step 4 of the main flow of events, it may happen that the system is unable to authenticate the user.
2. The system then notifies the user of the unsuccessful login attempt and presents the login page to the user.
3. Now the main flow is repeated from step number 2.

Alternate flow of events 2:
1. The user is only allowed 5 unsuccessful attempts before the system locks for a certain (undecided) period of time before a re-attempt is allowed.

Entry condition: The user has started the system
Exit condition: The user has been successfully logged in or has been notified of the reason for the unsuccessful attempt.

1.2: Recover password

Use case name: Recover password.
Participating actors: System user.
Flow of events:
1. The user does not remember the password to its account.

2. The user can click on the ‘forgot my password’ button to recover password.
3. The user is asked for email id associated with the account so that the recovery code can be sent to the user.
4. The user is presented with a screen asking to enter the recovery code that was sent. The code has a timed validity and the screen will return to login screen after a specified time
5. The user copies the recovery code from the email and pastes the code inside the recovery code box.
6. If the code is correct and not timed out, the system asks the user to enter new password.
7. The password is successfully changed after the user presses the accept button.
8. The system notifies the user of a successful password change and returns the user back to login screen.
Alternate flow of events 1:
1. At step 4 of main flow of events, if the user enters an invalid code/expired code for more than 2 times, the password recovery process is terminated and the user will be taken to login screen again.
Entry condition: The user has forgotten account password and wishes the recover the password.
Exit condition: The user successfully recovers account password or is notified of the reason for an unsuccessful attempt.

1.3: Change password
Use case name: Change password.
Participating actors: System user.
Flow of events:
1. The user is logged into its own profile.
2. The user can click on the ‘Change my password’ button to change password.
3. The user is asked for old password to verify the process.
4. If the old password is correct, the user is asked to enter the new password for the account.
5. The user should press confirm changes button to confirm the change in password,
6. The system notifies the user of a successful password change and returns the user back to login screen.

2. Use Cases for Student
2.1: Login into the Portal
Participating actors: Student.
Flow of events:
The user can login into the system by providing its loginid and password to the portal which are unique and provided by the college.
To login into the portal and take advantage of the services provided by it.

2.2: Change password
Participating actors: Student.
Flow of events:
Change the password after logging into the system.The user will be asked a security question if he/she forgets his/her password. In case, he/she is unable to do so, the user has to formally apply to the admin for the change of his/her password providing the necessary proof.
For security purposes, the user can choose his/her password.

2.3: View/change profile details
Participating actors: Student.
Flow of events:

The user can view or change some of his/her personal details like email id, contact details and address details. The profile will contain name, age,permanent address, parent's name, their address, their contact details, branch, year, semester, room alloted, hostel name and no. etc.
To update his/her profile and know his or her status.

2.4: Help

Participating actors: Student.
Flow of events:
DESC: Can get help through the help option which gives information about the different features provided by the system. A copy of user manual will also be provided in this section.
In case of any confusion, the user can solve the problem easily and better understand the functionality of the system.

2.5: Course Evaluation

Participating actors: Student.
Flow of events:
DESC:: At the end of the semester, a new page will be available for evaluation of a particular course for the students. The user can give evolution for a particular course by just filling the multiple choice questions and submitting.
This helps faculties to get a feedback for their course and improve their performance by introspecting and also it helps the administration in better allocation of the faculty.

2.6: Request for certificates

Participating actors: Student.
Flow of events:
DESC:The user can request for different types of certificates like No Objection Certificate, Character Certificate, Address Proof for Hostel etc. After filling the given format, the request will be send to the administrative department for verification of request. After the verification, the user can get his/her certificate.

2.7: Exam Schedule

Participating actors: Student.
Flow of events:
DESC:The user can view the exam schedule.
By using this feature, the user can prepare and plan his efforts in a better way.

2.8: Feedback/Complains

Participating actors: Student.
Flow of events:
DESC: In this form, one can lodge a complaint or give a feedback by selecting the domain: Academics, Administration, Hostel, Mess, Transportation. The head of each department will give response accordingly.
The user can get relieve from his/her problems without much effort or conflicts.

2.8: Fee Receipt Generation
Participating actors: Student.
Flow of events:
DESC: Student can generate fee receipt by giving the bank details in the system authenticated by the administrative head for the current and the previous semesters.
This feature makes the process move convenient, fast and less cumbersome.

3. Use Cases for Faculty Staff

3.1: Assignments and reading references
Participating actors: Faculty Staff.
Flow of events:
DESC: The user can upload the assignments on to the portal according to the year, batch and course.
The user will have a remote access to the assignments.

3.2: Exam Schedule
Participating actors: Faculty Staff.
Flow of events:
DESC: The user can view the exam schedule.
By using this feature, the user can prepare and plan his efforts in a better way.

3.3: Feedback/Complains
Participating actors: Faculty Staff.
Flow of events:
DESC: In this form, one can lodge a complaint or give a feedback by selecting the domain: Academics, Administration, Hostel, Mess, Transportation. The head of each department will give response accordingly.
The user can get relieve from his/her problems without much effort or conflicts.

3.4: Change password
Participating actors: Faculty Staff.
Flow of events:
DESC: Change the password after logging into the system. The user will be asked a security question if he/she forgets his/her password. In case, he/she is unable to do so, the user has to formally apply to the admin for the change of his/her password providing the necessary proof.
For security purposes, the user can choose his/her password

3.5: View/change profile details
Participating actors: Faculty Staff.
Flow of events:
DESC:The user can view or change some of his/her personal details like email id, contact details and address details. The profile will contain name, age, permanent address, parent’s name, their address, their contact details, branch, year, semester, room allotted, hostel name and no. etc.

3.6: Help

Participating actors: Faculty Staff.
Flow of events:
DESC: Can get help through the help option which gives information about the different features provided by the system. A copy of user manual will also be provided in this section.
In case of any confusion, the user can solve the problem easily and better understand the functionality of the system.

4. Use Cases for Professors

4.1: Add exam/quiz to the course (Grading criteria)

Use case name: Add exam/quiz to the course
Participating actors: Professor
Flow of events:
1. The professor is logged into the system.
2. The professor enters the 'grade it' tab.
3 The system opens the tools presenting the list of exams currently added to the course. The professor chooses to add a new grading criteria to the course by pressing 'add new criteria' button.
4. The system presents a form to the user asking for details about the grading criteria.
5. The professor enters the details about the criteria and presses submit button.
6. The system adds the new criteria to the course and returns to the list of criteria.

Entry condition: The professor is logged in to the system and is associated with the course.
Exit condition: The professor successfully adds the new grading criteria to the user or is given a appropriate notification about what went wrong with the process.

4.2: Add marks

Use case name: Add/Change marks.
Participating actors: Professor, TA
Flow of events:
1. The professor/TA is logged into the system.
2. The professor/TA opens the 'grade it' tab.
3. The system displays the list of criteria of evaluation that currently exist in the course.
4. The professor/TA press the 'Add marks' button along the criteria top add/edit marks.
5. The system presents the user with the name of all the students currently taking this course along with the marks added (if any).
6. The user can add the marks alongside each students' name.
7. The user presses the submit button to complete the process and save changes.
8. The system returns back to the list of criteria.

Entry condition: The professor/TA is logged in the system and is associated with the course.
Exit condition: The professor/TA successfully adds/edits the marks associated with grading criteria.

4.3: Calculate curve

Use case name: Calculate curve.
Participating actors: Professor
Flow of events:
1. The professor enters the 'curve it' tool.
2. If any previous curve calculation for the associated course exists, the system presents it to the user.

3. The user can now either press calculate curve or recalculate curve button to start the curving process.
4. The system presents a form to the user so that the user can enter curving parameters.
5. The user now presses the submit button and the system starts calculation of curve along the given parameters.
6. The system presents the new curve along with details of what the new grades for each student are.
7. The user presses the 'apply new grades' button to confirm the grading curve. The user also has the option of pressing cancel to keep the old grading curve.
Entry condition: The professor must be logged into the system and there must be at least one grading criteria defined.
Exit condition: The professor successfully calculates the grading criteria and saves it to the system or is given an appropriate message about what went wrong.

4.4: Share course documents

Use case name: Share course documents
Participating actors: Professor, TA
Flow of events:
1. The user logs in the system and enters the 'share it' tab.
2. The system presents an interface to the user with the facility to attach files and add notes.
3. The user adds appropriate documents and presses send button.
4. The system automatically adds all the students associated with the course to the mailing list and send an email to each student.
5. The system provides a confirmation to the user.
6. The system returns back to the user's home screen.
Entry condition: The user is logged in and is associated with at least one of the courses.
Exit condition: The user can successfully share files with students or is given a appropriate notification about where the process faltered.

4.5: Search for courses

Use case name: Search for courses.
Participating Actors: System User: Student
Flow of events:
1. The student first enters the course search tab.
2. The student is given option to search for courses by department or by name.
3. The student presses the search by department button.
4. The student is presented with the list of departments which offer courses.
5. After selecting the department, the student is presented the list of courses offered by the department.
6. The student can go back and forth and select different departments for its search.
Alternative flow of events:
1. The student first enters the course search tab.
2. The student is given option to search for courses by department or by name.
3. The student presses the search by name button.
4. The student types the name of the course in the search box and presses the search button.
5. The student is presented with the courses matching the search query.
6. The student can type a different search name and get other results.
Entry Condition:
Student must be logged in to the system.
Exit Condition:
The student is able to see the list of courses searched for.

4.6: Register for courses

Use case name: Register for courses.
Participating Actors: Student
Flow of events:
1. The student searches for courses either by department or by name and has a list of courses displayed.
2. The student selects a course from the list.
3. The system displays the course details to the student. The details of the course also includes the current capacity and the filled capacity of the class.
4. If the class has remaining capacity and the student is eligible to register, the register for class button will be enabled and student can press the button to register.
5. The request for the registration is processed and the student is registered for the course.
Alternate flow of events 1:
1. From step 4 of the main flow of events, if the class capacity is full, the register button will be disabled and the 'add to wait list button' will be enabled.
2. The student can add self to the wait list for the current course if the eligibility criteria is met by the student.
3. After pressing add to wait list button, the student is added to the wait list for the course.
Alternate flow of events 2:
1. From step 4 of the main flow of if the student is not eligible at all to register for class, the buttons for registration and waiting list will be disabled.
2. The student can now go to a different course where registration is possible.
Entry Condition:
Student must be logged in to the system.
Exit Condition:
The student has either being able to register for the course or the waiting list or is shown proper message regarding the eligibility criteria.

4.7: Dropping courses/ removing from wait list.

Use case name: Drop course
Participating Actors: System User: Student
Flow of events:
1. The student enters the 'My courses tab'.
2. The current courses on the registration and the waiting list of student is displayed to the student
3. The student can select the drop course checkbox for the courses which are supposed to be removed either from main registration or the waiting list the student is subscribed to.
4. The student now presses the save changes button to confirm dropping of courses.
5. The system presents a summary of changes to be made and asks the student to confirm the changes.
6. The student can press the confirm changes button and the changes will be applied to its profile.
7. The 'My courses tab' is displayed again to the student.
Alternative flow of events 1:
1. At step 4, the student can press cancel changes button.
2. The system will ignore all the checkbox for dropping courses and refresh the 'My courses tab' without applying any changes.
Alternative flow of events 2:
At step 6, the student can press cancel changes button.
The system will ignore all the selected courses for dropping and refresh the 'My courses tab' without applying any changes.

Entry Condition:
Student must be logged in to the system and inside the my courses tab
Exit Condition:
The changes requested by the student have been applied.

4.8: View grades

Use case name: View grades
Participating actors: Student
Flow of events:
1. The student enters the 'my grades' tab.
2. The student is shown the list of courses the student is currently registered for or was previously registered. The system also displays the current cumulative GPA for the student excluding the current courses taken.
3. The student clicks on the course for which grade is to be viewed.
4. The system displays all the grades for the particular course including any exam or homework grade. The grade display depends upon what grades have been posted by the associated professor.

Entry condition:
The student is logged in and is or was registered for at least one course
Exit condition:
The student is able to view the cumulative and course wise grades.

4.9: See course details

Use case name: View course details
Participating actors: Student
Flow of events:
1. The student enters the 'my courses' tab.
2. The student is shown the list of courses the student is registered for.
3. The student clicks on the course for which details are to be viewed.
4. The system displays all the course details to the student including professor details, TA details, syllabus, recommended books, etc.
5. The student can also download attachments from the course page that may have been made available by the professor teaching the course.

Entry condition:
The student is logged in and is registered for at least one course.
Exit condition:
The student is able to view the course details and view/download all the available course material.

4.10: View Student details

Use case name: View student details
Participating actors: Student
Flow of events:
1. The student enters the 'my details' tab.
2. The system shows the student the details about the student in terms of registration, course summary, grade summary etc. (Summary)

Entry condition:
The student is logged in.
Exit condition:
The student is able to view the details about self or an appropriate message indicating the error is shown.

4.11: Check eligibility to apply

Use case name: Check eligibility to apply
Participating Actors: Student
Flow of events:
1. The student enters the page containing the position description.
2. The student then has to check if he is eligible to apply for the position.
3. 'Apply' button then checks if the logged student has the required GPA to apply for a position.
4. If criteria matches then the Application form is displayed.

Alternative flow of events:
1. The student enters the page containing the position description.
2. The student then has to check if he is eligible to apply for the position.
3. 'Apply' button then checks if the logged student has the required GPA to apply for a position.
4. If criteria does not match then the application form should not be shown but a pop up message box should address 'criteria not matching'.

Entry Condition:
Student must have required GPA.
Exit Condition:
The student is able to view the application form if the eligibility criteria is satisfied else the pop up message box should be shown.

4.12: Enter Application Details

Use case name: Enter application details
Participating Actors: Student
Flow of events:
1. The student enters the page containing application form.
2. The student answers the various required details.
3. Then to save the data "confirm" button is pressed.
4. The confirmation overview page is then opened which shows all the data entered for verification.
5. If there is a need to change any entry 'Edit' can then be pressed to go back to the previous page.
6. If the student finds all entries to be correct, he can then press "Submit".

Entry Condition:
Student must be logged in and must be eligible to apply.
Exit Condition:
The student is able to complete the application for open positions.

4.13: Scale Students

Use case name: Scale students
Participating Actors: Nil (Initiated by the system)
Flow of events:
1. Prior to the student entering the details in the application form and the data is saved in the database.
2. Then the system automatically computes the skillset into points/ flags and generates a score.
3. This set of flags and the score is saved into another database table.
4. When the match students is initiated this data is used for matching the requirements.
5. The appropriate matching list is then returned to the user.

Entry Condition:
The student should have filled the application form.
Exit Condition:

The flags and the score is saved in another database table.

4.14: Match Requirements

Use case name: Match requirements
Participating Actors: Professor, department admin
Flow of events:
1. The professor/ department logs into the system.
2. Then by clicking on ‘add position’ will direct them to the appropriate page.
3. The professor/ department enters the skillset required for the position.
4. Then presses ‘Match’ to generate a list of students who have similar requirements.

Entry Condition:
Professor/ department must be logged in to add an open position.
Exit Condition:
The Professor/ department is able to see the list of matching students.

4.15: Intimate Students

Use case name: Intimate students
Participating Actors: Professor/ department admin
Flow of events:
1. The professor/ department logs into the system.
2. Then the professor/department performs the ‘match requirements’ as explained in the previous use case.
3. The professor/ department can select the number of people he wishes to call for an interview.
4. Accordingly a mail is sent to them with further details.
5. A summary report email is then sent to the professor too.

Entry Condition:
Professor/ department must be logged in to add an open position.
Exit Condition:
The students and the professor/ department are able to receive a mail.

4.16: Add a course

Use case name: Add a course
Participating actors: Department admin
Flow of events:
1. The department admin enters the manage courses tab.
2. The system displays the current courses that are under the department.
3. The department admin presses the ‘add a new course’ button.
4. The system produces a form asking for details about the course.
5. The department admin can now add all the details and press the apply button
6. The system now asks for confirmation from the user to add the course.
7. The department admin now can press the confirmation button to add this new course to the course list.
8. The system adds the new course and presents the courses page to the department admin.

Entry condition:
The user is logged in as a department admin.
Exit condition:
The department admin is able to add new course or is given an appropriate notification about where the process faltered.

4.17: Remove a course
Use case name: Remove a course
Participating actors: Department admin
Flow of events:
1. The department admin enters the manage courses tab.
2. The system displays the current courses that are under the department.
3. The department admin can now select a course from the list that is to be deleted.
4. The system now displays all the information related to the course clicked.
5. The department admin now presses the remove course button.
6. The system asks for confirmation from the user to delete the course.
8. The system removes the course and presents the courses page to the department admin.
Entry condition:
The user is logged in as a department admin.
Exit condition:
The department admin is able to remove course or is given an appropriate notification about where the process faltered.

4.18: Add a professor
Use case name: Add a professor
Participating actors: Department admin
Flow of events:
1. The department admin enters the 'manage teaching faculty' tab.
2. The system displays the current professors that are under the department.
3. The department admin presses the 'add a new professor' button.
4. The system produces a form asking for details about the professor.
5. The department admin can now add all the details and press the apply button.
6. The system now asks for confirmation from the user to add the new professor.
7. The department admin now can press the confirmation button to add this new professor to the professors list.
8. The system adds the new professor and presents the current professors page to the department admin.
Entry condition:
The user is logged in as a department admin.
Exit condition:
The department admin is able to add new professor or is given an appropriate notification about where the process faltered.

4.19: Add a student
Use case name: Add a student
Participating actors: Department admin
Flow of events:
1. The department admin enters the 'manage students' tab.
2. The system displays the current students that are under the department.
3. The department admin presses the 'add a new student' button.
4. The system produces a form asking for details about the student.
5. The department admin can now add all the details and press the apply button.
6. The system now asks for confirmation from the user to add the new student.

7. The department admin now can press the confirmation button to add this new student to the students list.
8. The system adds the new student and presents the current students page to the department admin.
Entry condition:
The user is logged in as a department admin.
Exit condition:
The department admin is able to add new student or is given an appropriate notification about where the process faltered.

4.20: Remove a professor/student

Use case name: Remove a professor/student
Participating actors: Department admin
Flow of events:
1. The department admin enters the ‘manage professors/students tab’.
2. The system displays the current professors/students that are under the department.
3. The department admin can now select a professor/student from the list that is to be deleted
4. The system now displays all the information related to the professor/student clicked.
5. The department admin now presses the remove professor/student button.
6. The system asks for confirmation from the user to delete the professor/student.
8. The system removes the professor/student and presents the current professors/students page to the department admin.
Entry condition:
The user is logged in as a department admin.
Exit condition:
The department admin is able to remove professor/student or is given an appropriate notification about where the process faltered.

Table 2. User experience form

Part 1: Usefulness
1. Is it easy to create and add a new model to a collaboration?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
2. Is it easy to add a new element to a model?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
3. Is it easy to delete an element of a model?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
4. Is it easy to modify an element in a model?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
5. Is the dashboard of the CoMPers platform user-friendly?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
Part 2.1: Teamwork
6. Is the platform appropriate for modeling with a team?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
7. Is the platform appropriate to propagate changes to others?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
8. Is an appropriate approach used to inform users of changes applied by collaborators?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
9. Is it easy to see the history of changes and reports of conflict?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
10. Would you recommend your friends use this platform for collaborative modeling?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
Part 2.2: Customization
11. Is it possible to customize the method of publishing changes for each model?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
12. Is it possible to customize the method of receiving changes for each model?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
13. Is it possible to appropriately personalize the change propagation in this platform?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
14. Is it possible to appropriately customize the conflict management approach?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea
15. Is it possible to customize collaboration on this platform?
☐ Strongly disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly agree ☐ No idea