CSE 6329 – Project Phase 2

This project will be a simulation of industry work.

This is a letter from your Project Manager. Please read it carefully and do the needful.

“

Hello Team,

Well done on your previous assignment! I’m impressed. Now it’s time to give you another assignment. This time you will use N-Process Model and the methodology for reengineering.

Our clients were happy with all the modifications your team did in the last assignment and hence they have requested more features to be added and modified.

Here is the list of features what they want for their appointment scheduling application.

- Implement CAPTCHA (spelling captcha) while making a new account to verify if the user is human or not.
- Forgot password option to change password. The user should receive an email providing a link to change their password.
- Lock the user’s account if the password is entered incorrectly after three attempts.
- Students get text message regarding the appointment before an hour to remind.
- Only show appointments for the current month and the next.
- If all slots are reserved for the day, the students shall be able to add their name and email address to the wait-list.
- Students can add the appointment time and date to their outlook calendar.
- Students should be able to customize the purpose of appointment regardless of adding, dropping, swapping classes.
- A notification feature that indicates if Advisor is available / present in University or not.
- Implement waiting list if all the time slots are full. Show a separate page on website to show the waitlist.
- Student should cancel the appointment if he/she is not able to attend. Failing to do that advisor can add the student to defaulter list and student will be notified (choose any method of notification: text message, email, notification in student account etc.) with $20 penalty fees.
- If there are any vacant slots available students could get that and notify student about the default advising session.
- Students should not be allowed to make multiple appointments on the same day with same professor
- Advisors can send a group email to all students who are registered under him for appointment

- Restricting a student from taking multiple appointments with same professor on the same day or with different professor on same day, same time.
- Student can now manage their profile that includes their photo (that they can upload), contact information, which course they are enrolled in and a summary of all past and upcoming appointments with the advisor.
- Student should be limited to make one appointment a day and 2 in a week.
- Advisor can send group email to every student enrolled for advising on a particular day if he cancels the appointment that day.
- Advisor can (color) label the appointments like urgent, general, etc.
- Feature to ask a Poll
- A link to UTA academic calendar
- Online advising if they can't meet physically then student should have ask questions on the portal. This can be like an online forum accessible by every UTA student.
- When the student forgets password then there should be some security questions to retrieve the account. Security questions should be asked while creating account.
- When students cancel their scheduled appointment, the application shall check the wait list and send an email notification to all student in the wait-list for that day informing them of the new availability.

Following is the list of deliverables I want from your team

1. PAL (Project artifacts list)
2. Re-write the above enhancement requirements as *formal set of requirements each* with a unique ID. Also group them accordingly.
3. Identify/Derive Use Cases from the formal requirements. The final document should consist of
    a. Formal requirements, derived use cases for each requirement
    b. Label the use cases as R (reuse), M (modified), N (new)
    c. Prioritize each use case (Critical, Major, Minor)
    d. Effort estimates (Fibonacci No)
    e. Allocated Release No
    f. Allocated iteration No
4. Domain Model (Optional but highly recommended to understand requirements)
5. Design Class Diagrams (clearly mark all modifications)
6. Design Sequence Diagrams (clearly mark all modifications)
7. Test Cases with traceability to each requirement (How you plan to test each requirement, input, output etc)
8. Test Case Report (Status of each test: Blocked, Pass, Fail, did not implement, etc)
9. Readme file (User document with step by step instructions how to set up the application)
10. Trace all around with Traceability matrix (Trace Id, tag corresponding to file, method, requirement id, test case, code etc.)

Looking forward to see brilliant work from your team. Thank you. “