

Assignment 1 COEN 6311 Summer 2024

Group Size of Two or Three

This assignment is defined within the context of DepartmentDirect. The context begins with software specification and continues the design and development of subtask within the scenario “subscribe to future updates and publish event notifications”.

The Example Prompt

DepartmentDirect is accessible to future applicants to receive events notifications, news updates and post questions and receive spontaneous response. Questions can be of different categories, such as how to navigate the web resources on the school or department websites; course sequences and pre-requisitions; career development resources and so on. Each department can have its own instance such as DepartmentDirect – ECE or DepartmentDirect – CSE that allows each department to customize the content given each department’s degree programs. Each department’s degree program directors and/or associated supporting staff members or admin officers can access DepartmentDirect for updating contents and post events. DepartmentDirect can be the form of Mobile App, or a Desktop App, or a Web App or a Chatbot.

DepartmentDirect include multiple scenarios. For example, the scenario of questions on course prerequisite (S1); the scenarios of questions on degree types, similarity and differences (S2); the scenarios of subscribe to future updates and publish event notifications (S3); the scenarios of user account management (S4) and more. Further features of the system can generate extra scenarios. The source of answers is from the content of Concordia website. Each scenario involves the interactions from both the registered future applicants and the department’s authorized faculties or staff. For each scenario, the business process can be fine defined starting from the entry page followed by the operations, activities and states of the systems.

The Software Specification. The tasks of software specification aim to identify the needs and requirements from the users of the DepartmentDirect; define the core features of the systems; for reach feature, further develop the user stories and assign priorities user stories. The user stories are used to follow the scrum practices with issues created in project management systems with roles, deadlines and states assigned to each issue and begin the agile software development.

Please generate the prompts using COSTAR framework for output epics and user story definition for the scenario “the scenarios of subscribe to future updates and publish event notifications”.

Generated prompts from Copilot

Epic: Subscribe to Future Updates and Publish Event Notifications

Context

Future applicants need a way to stay informed about department events and updates.
Departments need a streamlined method to publish notifications and updates.

Objective

Enable future applicants to subscribe to updates and receive notifications about events, and allow departments to publish these notifications efficiently.

Structure

- **Feature:** Subscription to updates
- **Feature:** Event notification publishing
- **Feature:** Notification management

User Stories

User Story 1: Subscribe to Updates

- **Context:** Future applicants want to stay informed about department events and updates.
- **Objective:** Allow users to subscribe to updates.
- **Structure:** Subscription form, confirmation, and management.
- **Tasks:**
 - Design a subscription form.
 - Implement email verification.
 - Create a subscription management interface.
- **Actions:**
 - As a future applicant, I want to subscribe to department updates so that I can stay informed about events and news.
 - As a future applicant, I want to manage my subscription preferences so that I can control the type of updates I receive.
- **Results:**
 - Users can subscribe to updates.
 - Users receive a confirmation email.
 - Users can manage their subscription preferences.

User Story 2: Publish Event Notifications

- **Context:** Departments need to inform future applicants about upcoming events.
- **Objective:** Allow departments to publish event notifications.
- **Structure:** Event creation, notification scheduling, and distribution.
- **Tasks:**
 - Design an event creation interface.
 - Implement notification scheduling.
 - Create a distribution system for notifications.
- **Actions:**
 - As a department admin, I want to create and schedule event notifications so that future applicants are informed about upcoming events.
 - As a department admin, I want to manage published notifications so that I can update or cancel events if needed.
- **Results:**

- Departments can create and schedule event notifications.
- Notifications are sent to subscribed users.
- Admins can manage existing notifications.

User Story 3: Notification Management

- **Context:** Both users and departments need to manage notifications effectively.
- **Objective:** Provide interfaces for managing notifications.
- **Structure:** User notification settings, admin notification dashboard.
- **Tasks:**
 - Design user notification settings interface.
 - Implement admin notification dashboard.
 - Create notification history logs.
- **Actions:**
 - As a future applicant, I want to view my notification history so that I can keep track of past updates.
 - As a department admin, I want to view and manage all sent notifications so that I can ensure accurate communication.
- **Results:**
 - Users can view and manage their notification history.
 - Admins have a dashboard to manage all notifications.
 - Notification history is logged for reference.

Generated prompts from ChatGPT

Epics and User Stories

Epic 1: Subscription Management

- As a future applicant, I want to subscribe to updates so that I can receive notifications about events and news.
 - **User Story 1.1:** As a future applicant, I want to see a subscription form on the homepage so that I can sign up for updates easily.
 - **User Story 1.2:** As a future applicant, I want to choose the types of updates I receive (e.g., event notifications, news updates) so that I only get relevant information.
 - **User Story 1.3:** As a future applicant, I want to manage my subscription preferences so that I can update my choices or unsubscribe if needed.

Epic 2: Event Notification Publishing

- As a department staff member, I want to publish event notifications so that future applicants are informed about upcoming events.
 - **User Story 2.1:** As a department staff member, I want to create and schedule event notifications so that they are automatically sent to subscribers at the appropriate time.
 - **User Story 2.2:** As a department staff member, I want to customize the content of event notifications so that they are specific to my department's events.
 - **User Story 2.3:** As a department staff member, I want to view the list of scheduled notifications so that I can manage and update them as needed.

Prioritization

1. **High Priority:** User Story 1.1, User Story 1.2, User Story 2.1
2. **Medium Priority:** User Story 1.3, User Story 2.2
3. **Low Priority:** User Story 2.3

Prompts for Creating Epics and User Stories

Epic 1: Subscription Management

- **Context:** "You are defining the features for subscription management in DepartmentDirect."
- **Objective:** "Create epics and user stories that allow future applicants to subscribe to updates and manage their preferences."
- **Structure:** "Break down the subscription process into specific tasks: viewing the subscription form, selecting update types, managing preferences."
- **Tools:** "Project management tools like Jira or Trello."
- **Audience:** "Future applicants who will subscribe to updates."
- **Relevance:** "Ensuring users can easily subscribe to relevant updates will improve engagement with DepartmentDirect."

Epic 2: Event Notification Publishing

- **Context:** "You are defining the features for publishing event notifications in DepartmentDirect."
- **Objective:** "Create epics and user stories that allow department staff to publish and manage event notifications."
- **Structure:** "Break down the notification process into specific tasks: creating and scheduling notifications, customizing content, managing scheduled notifications."
- **Tools:** "Project management tools like Jira or Trello."
- **Audience:** "Department staff who will publish notifications."
- **Relevance:** "Ensuring staff can efficiently publish event notifications will keep future applicants informed and engaged."

Example Prompts

1. **Subscription Management:**
 - "Describe the process a future applicant would follow to subscribe to updates on DepartmentDirect."
 - "List the types of updates a future applicant might want to receive and how they would select these preferences."
 - "Explain how a future applicant would manage their subscription preferences, including updating choices or unsubscribing."
2. **Event Notification Publishing:**
 - "Outline the steps a department staff member would take to create and schedule an event notification on DepartmentDirect."
 - "Describe how a department staff member would customize the content of an event notification."
 - "Explain how a department staff member would manage the list of scheduled notifications, including making updates or changes."

1. Above are examples generated by AI agents. However, they may not satisfy the criteria and constraints of the project scope. Please define the rubric following the guideline [FOTL-Rubric-Roadmap.pdf \(syr.edu\)](https://www.syr.edu/fotl-rubric-roadmap.pdf) . An example rubric is illustrated in the appendix. Please consider the aspects, such as
 - Are they independent from each other?
 - Are they clear to define tests and criteria to verify them?
 - Are they properly defined without over the scope? (They state what is needed but not how, not specific details)
2. Use the rubrics to score above two sources generated by AIs. Summarize the major limitations of each source by at least three aspects.
3. Provide revision and create your final epics and user stories following the format below. Score your user stories and compare with the AI generated user stories in a table.

As a customer, I want to _____ so
 that _____

User story	Agent 1 definition	Agent 2 definition	Your final definition
Content			
Norm Rubric Score			
Norm Cosine Similarity			

4. Select one user story from the ones you identified above and identify at least three conditions of satisfaction. Make one of them a non-functional requirement.

5. For the selected user story, break it down into tasks and assign the tasks into the project management system. Select two tasks for implementation and integrate the task development with code IDE, repository and project management system. Please demonstrate the unit testing is passed before the commitment to code repository and project management system reflects the change of states. The states should show the time tracking of the development of the two tasks.

6. Reflect on the user story definition. If there is deviation from the development and testing of two tasks, please revise the user stories. If any need to update the rubrics, please make the change. For both cases, please update the table of scores.

User story	Agent 1 definition	Agent 2 definition	Your final definition
Content			
Norm Rubric Score			
Norm Cosine Similarity			

7. Submit the report to address item 1-6 and provide the screenshots of project management system and links to github repo for this assignment. The template report is from <https://www.ieee.org/conferences/publishing/templates.html> or [IEEE Conference Template - Overleaf, Online LaTeX Editor](#). Mis-formatted documents will result in not being marked.

Appendix : Example Rubric For Evaluating User Stories

Criteria	Excellent (5 points)	Good (4 points)	Satisfactory (3 points)	Needs Improvement (2 points)	Unsatisfactory (1 point)
Clarity and Completeness	The user story is clear, concise, and complete. It follows the standard format. All necessary details are included.	The user story is mostly clear and complete, with minor omissions or ambiguities.	The user story is understandable but lacks some details or clarity.	The user story is vague and incomplete, missing key details.	The user story is unclear and incomplete, making it difficult to understand the requirements.
Relevance and Alignment with Objectives	The user story is highly relevant and aligns perfectly with the project objectives and user needs.	The user story is relevant and aligns well with the project objectives and user needs.	The user story is somewhat relevant but may not fully align with the project objectives.	The user story has limited relevance and does not clearly align with the project objectives.	The user story is irrelevant and does not align with the project objectives.
Acceptance Criteria	The user story includes clear, specific, and testable acceptance criteria that cover all aspects of the story.	The user story includes acceptance criteria that are mostly clear and testable, with minor omissions.	The user story includes acceptance criteria, but they are somewhat vague or incomplete.	The user story includes acceptance criteria, but they are unclear or not testable.	The user story lacks acceptance criteria or they are very vague and untestable.
Feasibility and Scope	The user story is feasible within the given timeframe and scope. It is	The user story is mostly feasible and appropriately sized, with minor	The user story is somewhat feasible but may be too large or complex for a	The user story is difficult to implement within the given	The user story is not feasible within the given timeframe and scope.

	appropriately sized for a single iteration.	adjustments needed.	single iteration.	timeframe and scope.	
Prioritization and Value	The user story is well-prioritized and provides significant value to the project and users.	The user story is appropriately prioritized and provides good value.	The user story is somewhat prioritized and provides moderate value.	The user story is poorly prioritized and provides limited value.	The user story is not prioritized and provides little to no value.