Software Requirements Specification (SRS)
Online Multiple-choice Exam Platform (OMEP)

Authors: Le Hoang
Customer:
Instructor: Prof. Nguyen Nhat Hai

I. Introduction

1.1 Purpose

Defining and describing the functions and specifications of the Online Multiple-choice Exam Platform (OMEP) is the primary goal of this Software Requirements Specification (SRS).This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality as specified by our customer. The intended audience of this document is our primary Book E-Commerce System customer: the ITSS Management instructor Prof. Nguyen Nhat Hai, the ITSS Management Group 1 members, as well as the other students attending ITSS Management that will require access to such documentation.

1.2 Scope

- The software system being produced is called Online Multiple-choice Exam Platform or OMEP.
- It is being produced for a customer interested in making a reliable online examination platform.
    - No need for separated exam location
    - Easy exam management
    - Automatic scoring system
    - Digital exam database
    - Easy examinee/ examiner management
    - High-end security
    - Require only simple web browser
- The system will be run on a central server with each user having a remote user interface through a web browser to interact with it.
- The Online Multiple-choice Exam Platform only allow admin to create user account. Authentication are available to all users.
- The user has the option to become an examinee, an examiner or an admin.
    - Examiners will be provided with tools to give scores manually, view exam reports, view progress of students in an ongoing exam, create, read, update, delete exams. They can also add or remove examinees from an exam, set up rules for scoring, and manage questions.
    - Examinees will be allowed to search for exams, take exams and view their progress records.
    - Admin can create, read, update and delete accounts, including examiner, examinee and admin accounts. Admin can also see system logs and monitor

systems through an admin dashboard. Both admin and Examiner can view Exam logs.

- The system also allows retrieving lost passwords using Gmail or Outlook.

1.3 Definitions, acronyms, and abbreviations

OMEP	Online Multi-choice Exam Platform
Admin	A user who creates an account and manages examiner, examinee and system function
Examiner	A user who creates and manage exams
Examinee	A user who takes an exam
CRUD	Create, Read, Update, Delete
Exam	A set of multichoice questions created by an examiner and taken by examinee
Score	A value inferred from a taken exam based examinee’s choice
SRS	Software Requirements Specification

1.4 References

1. "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.

1.5 Overview

The rest of the SRS contains:

- Overall description
    - Product perspective
    - Product functions
    - User characteristics
    - Constraints
    - Assumptions and dependencies
- Specific requirements
- Appendices
- Index

II. Overall description

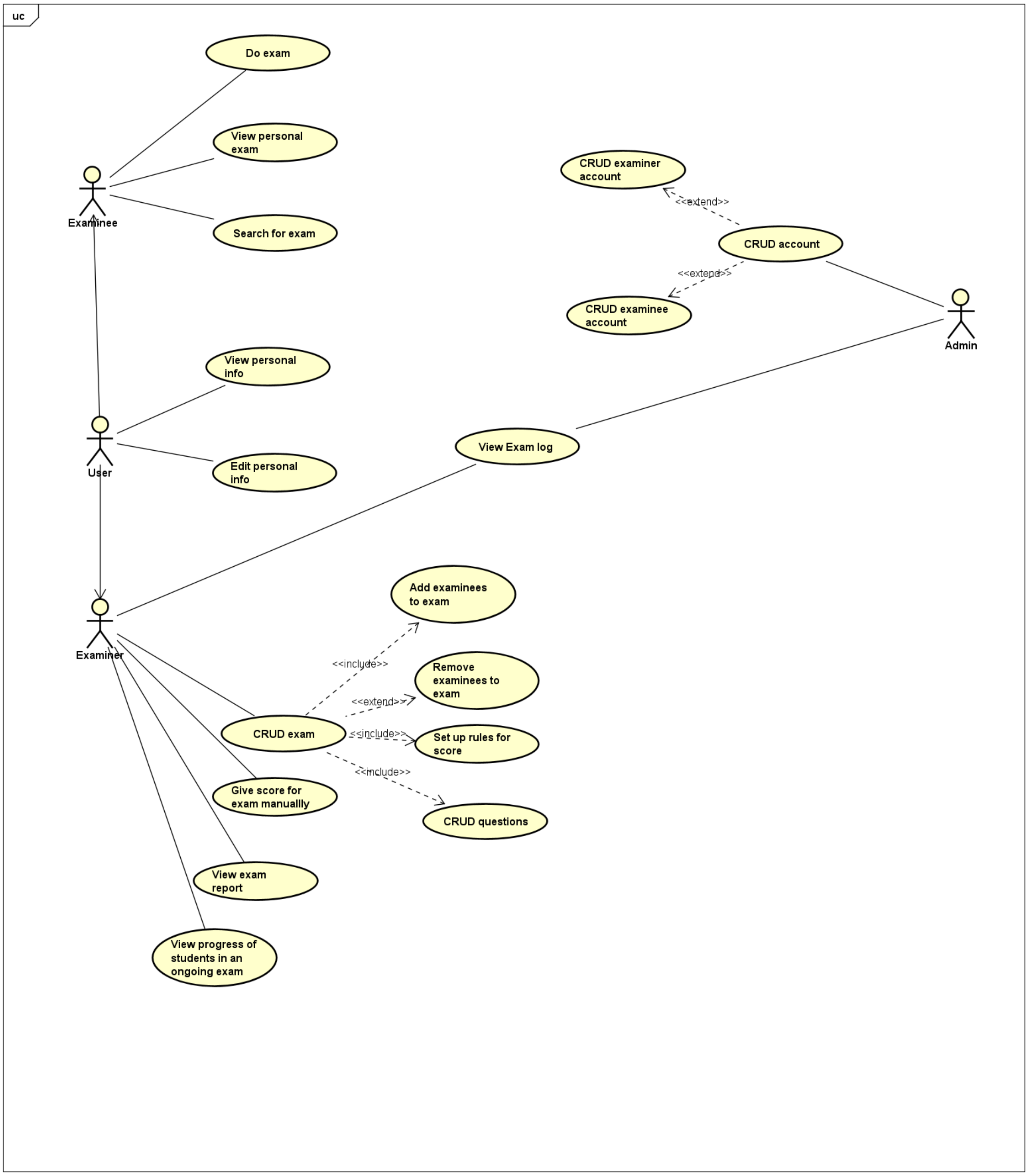

Figure1. Usecase Diagram

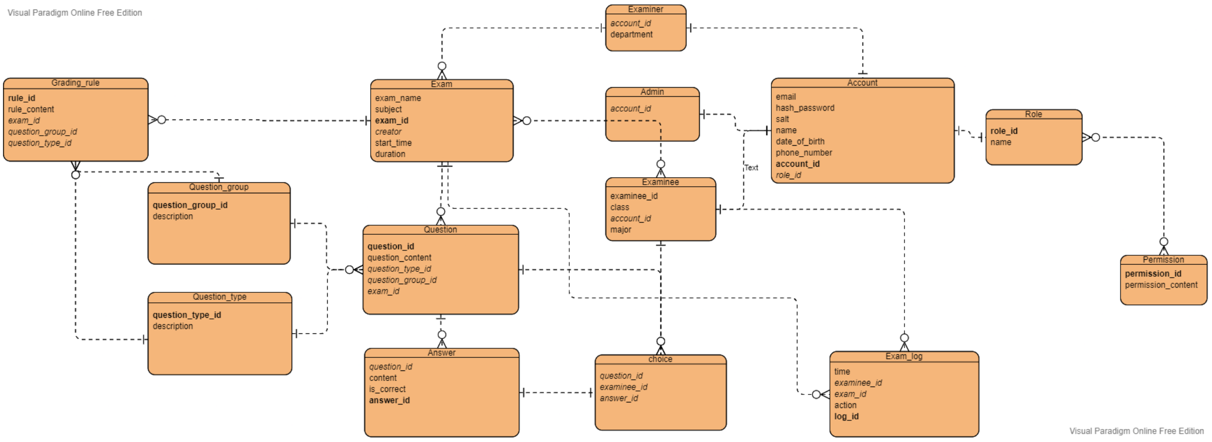

Figure 2. ERD Diagram

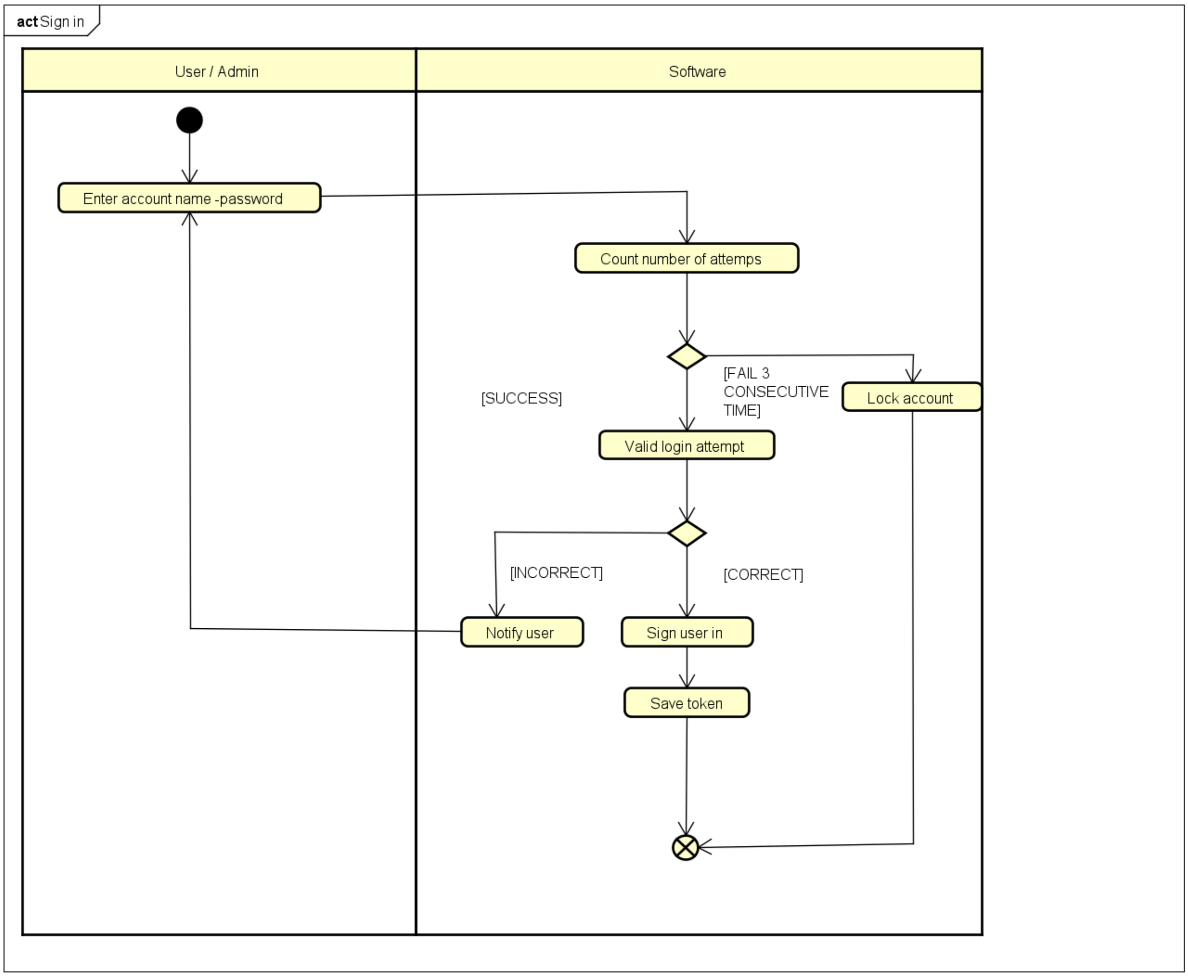

Figure3. Sign In Activity Diagram

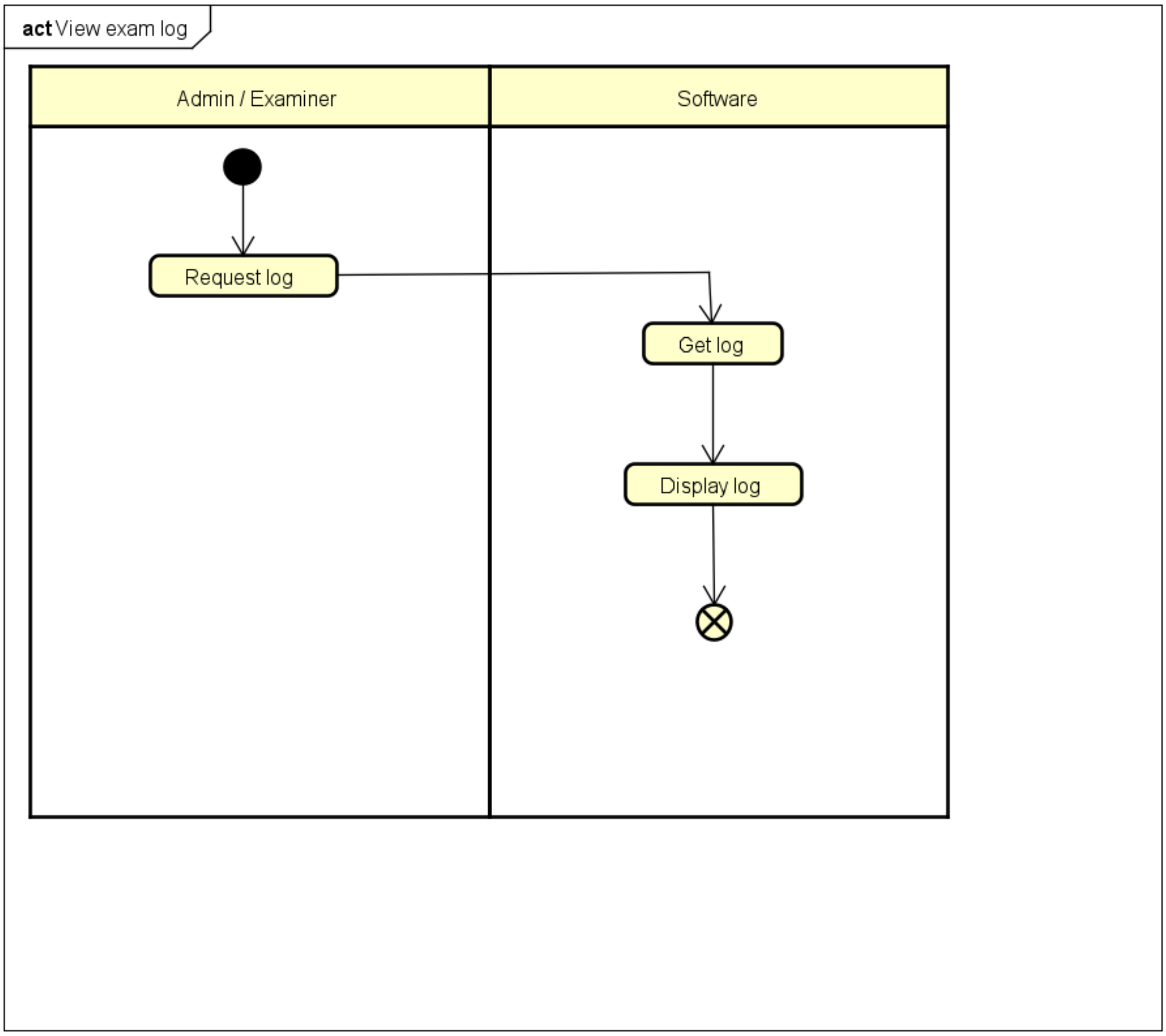

Figure4. View Exam Log

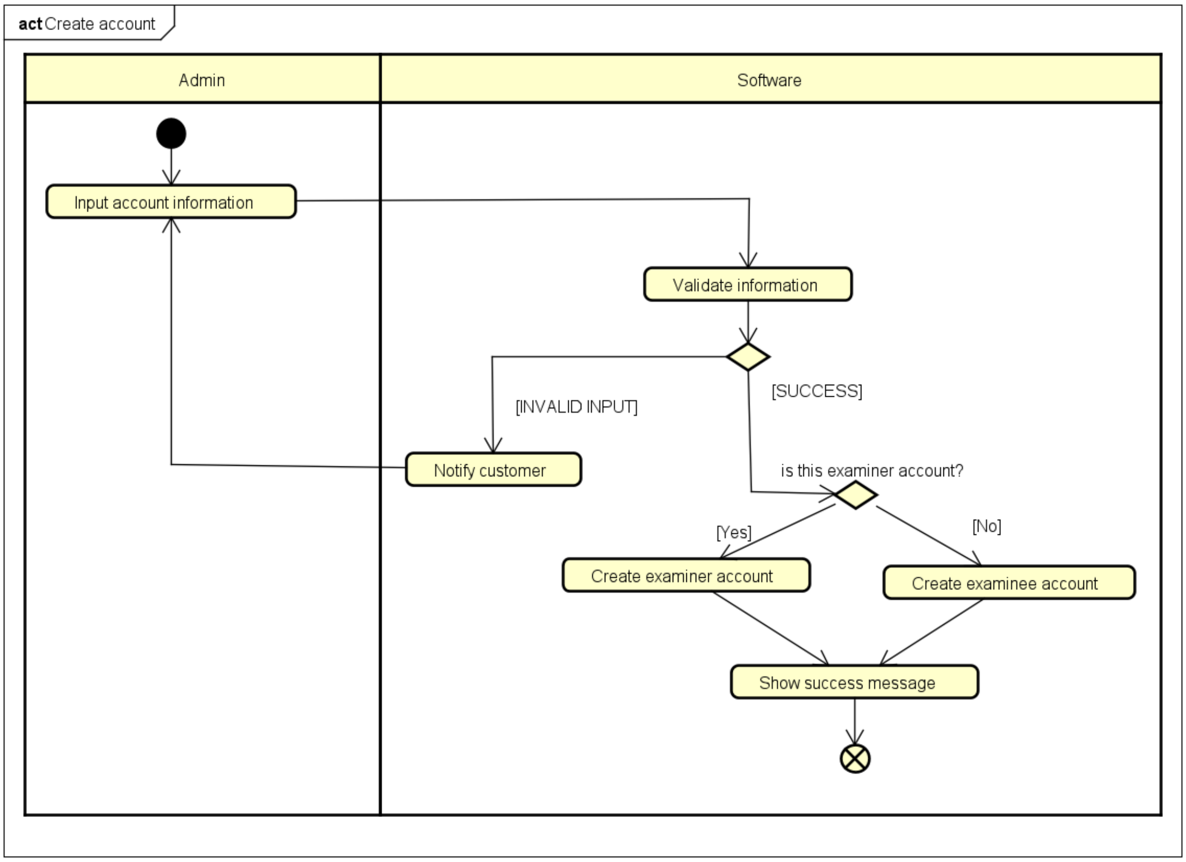

Figure5. Create Account Activity Diagram

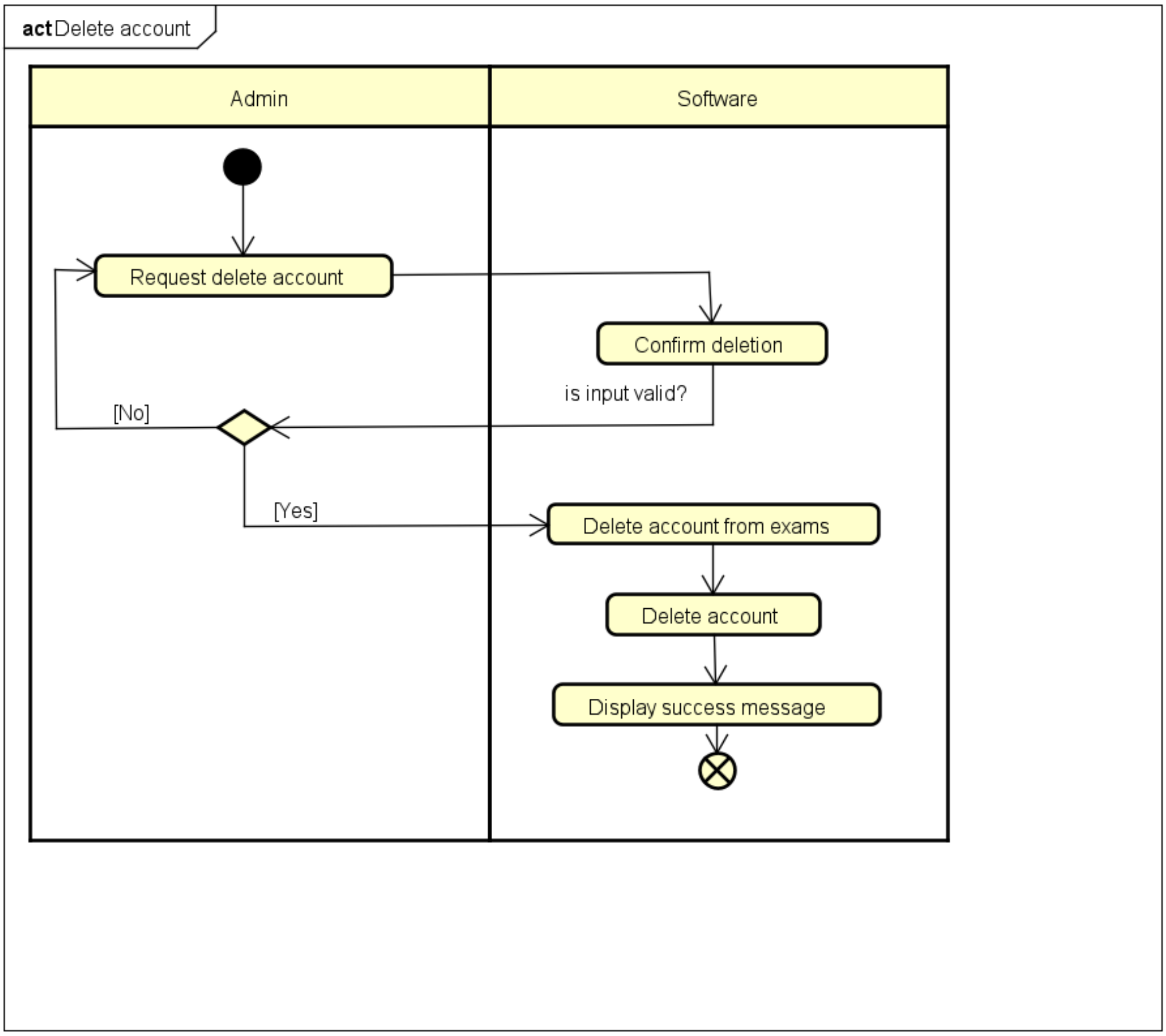

Figure6. Delete Account Activity Diagram

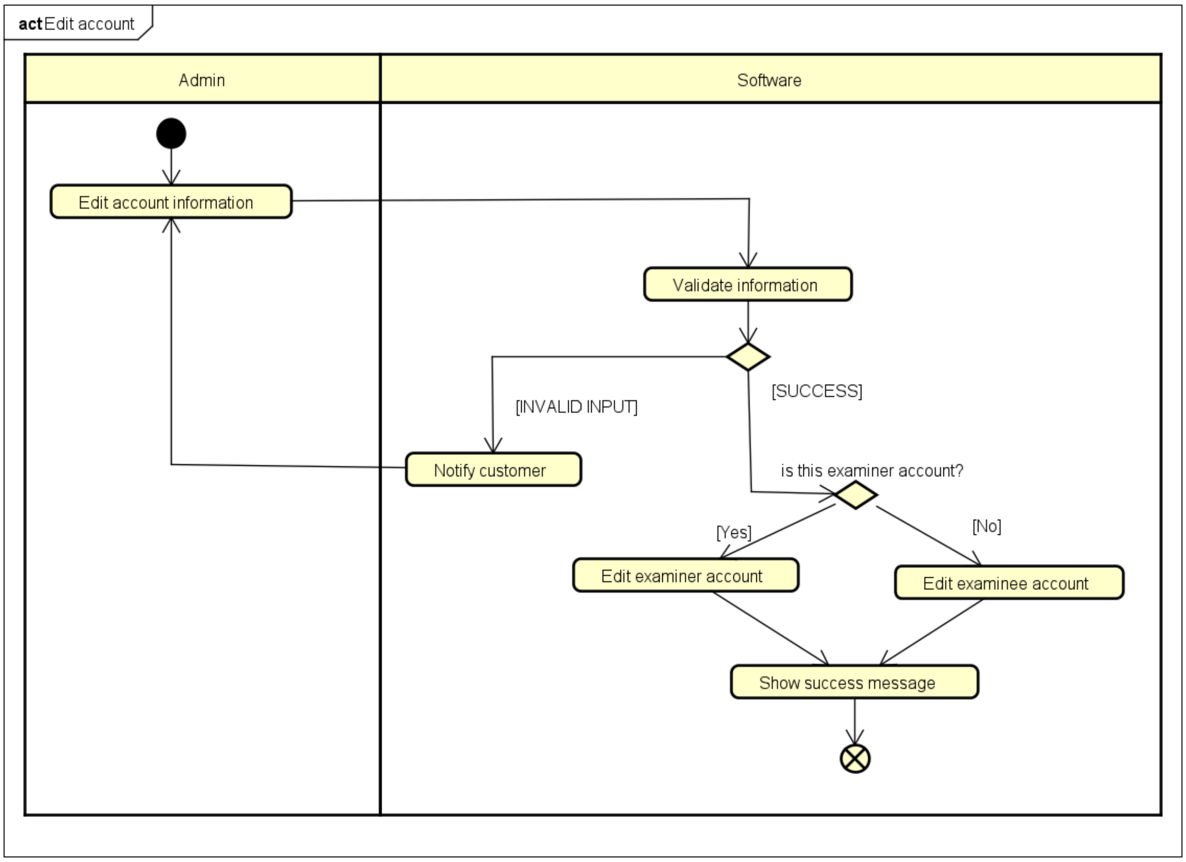

Figure7. Edit Account Activity Diagram

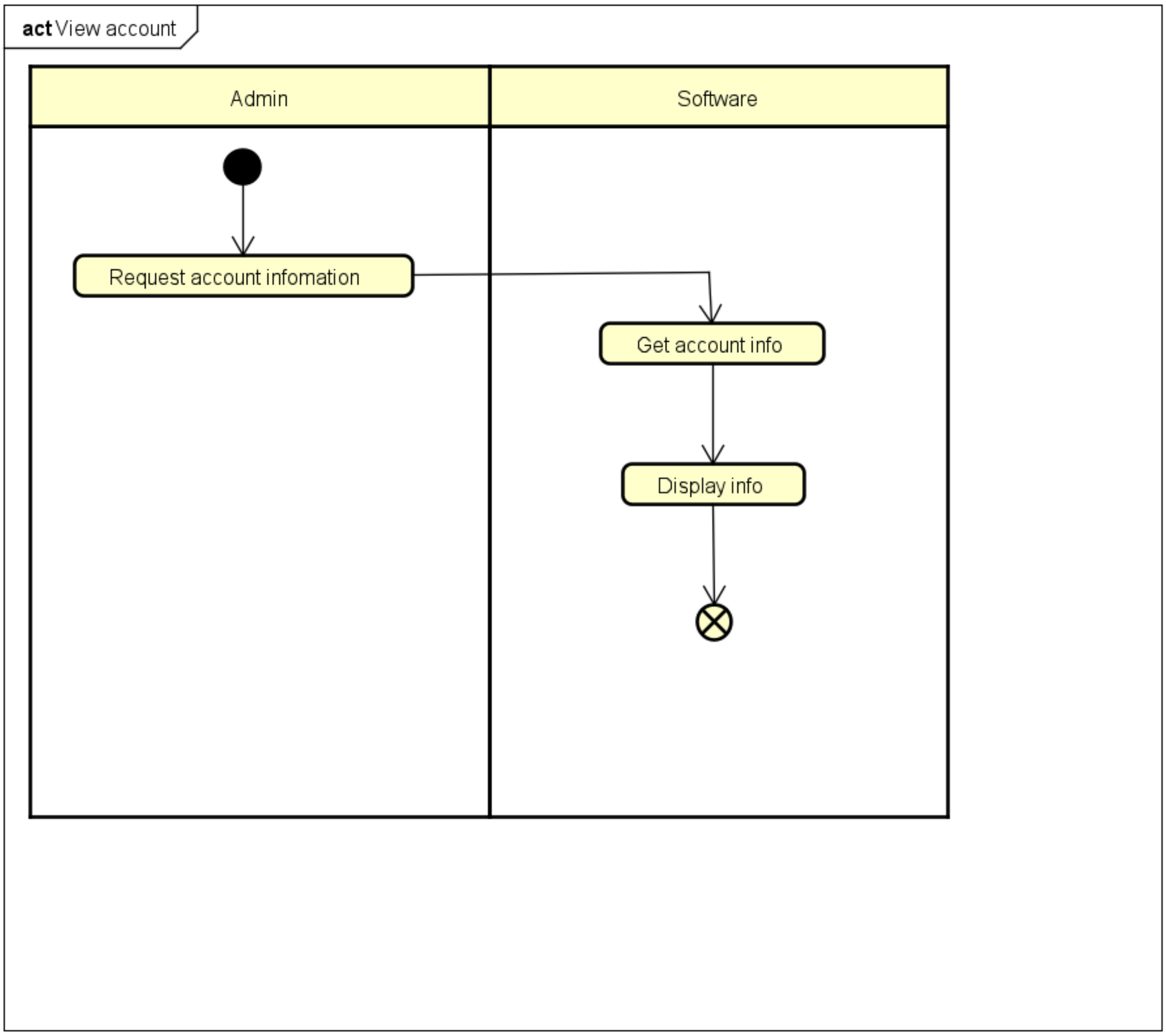

Figure8. View Account Activity Diagram

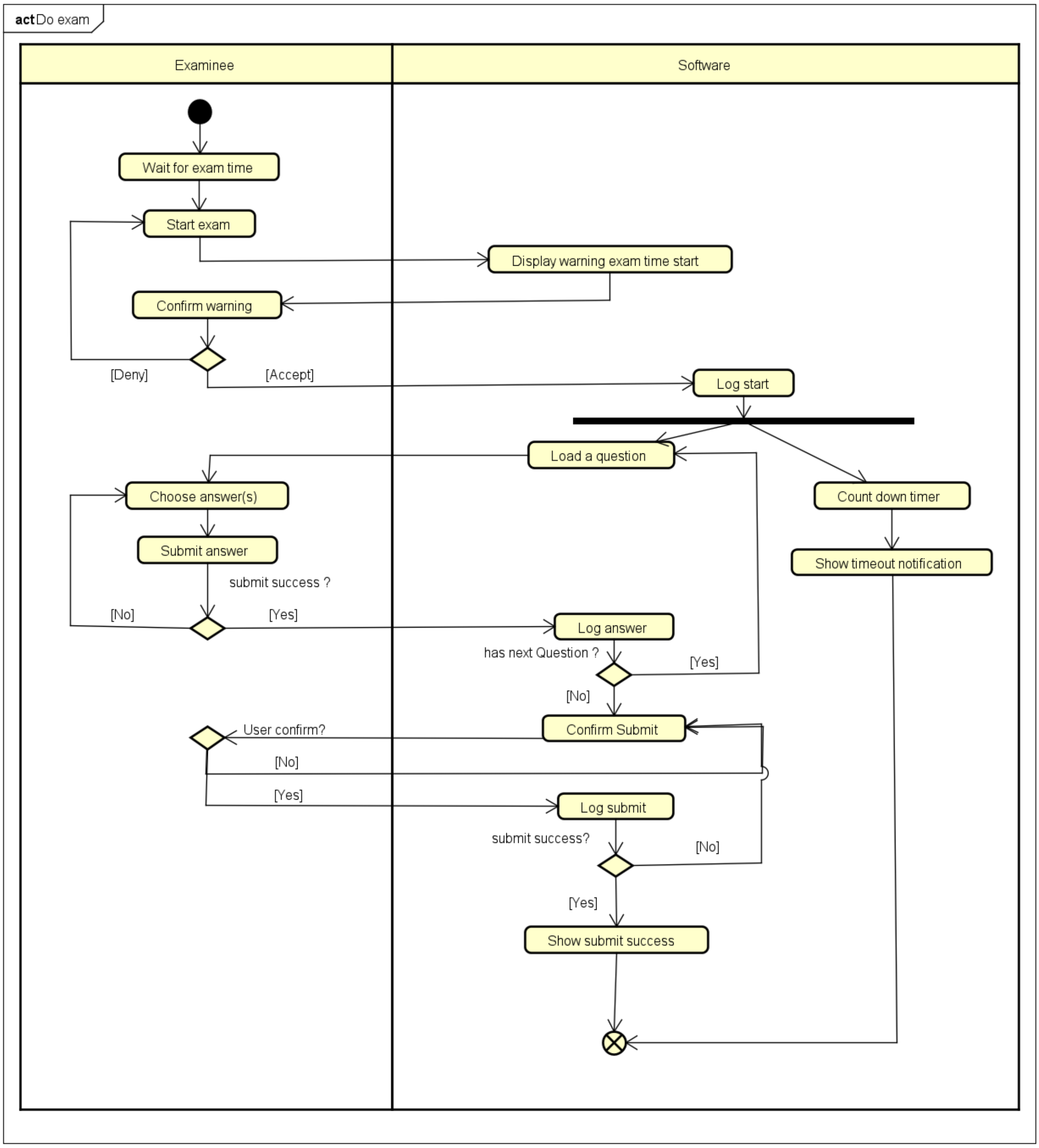

Figure9. Do Exam Activity Diagram

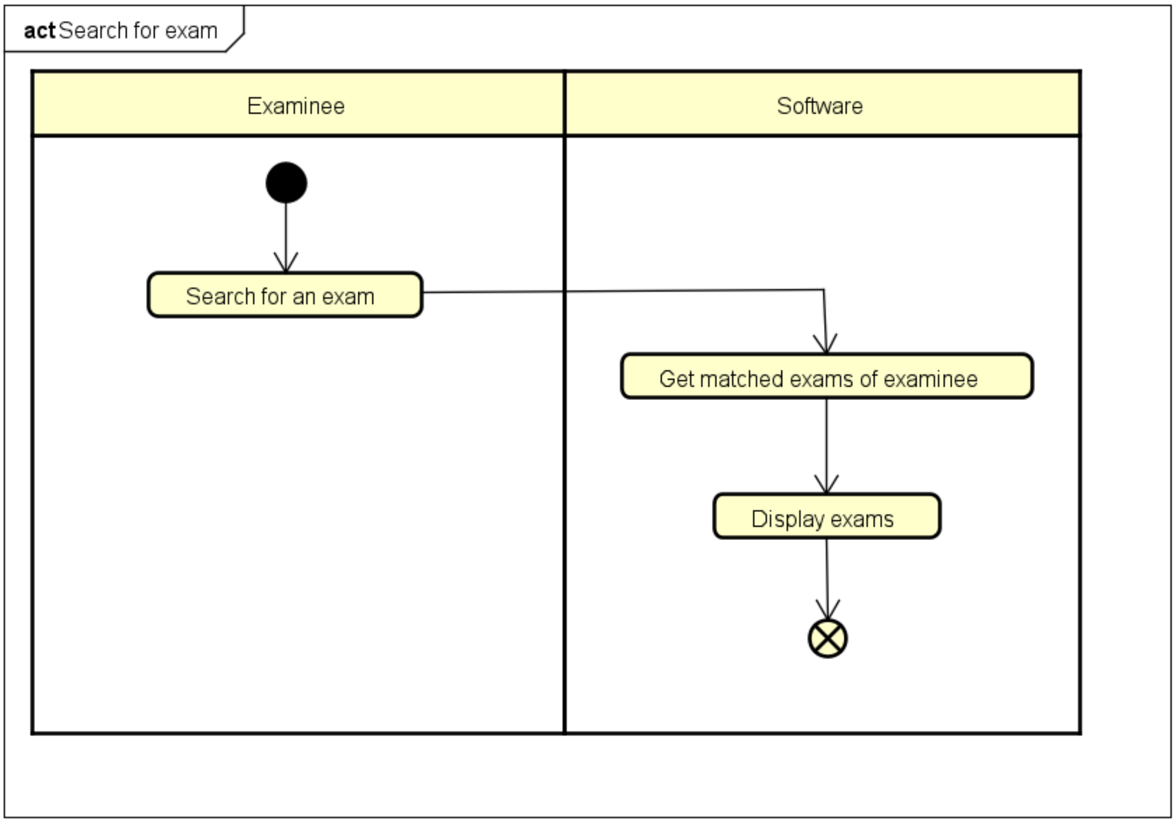

Figure10. Search For Exam Activity Diagram

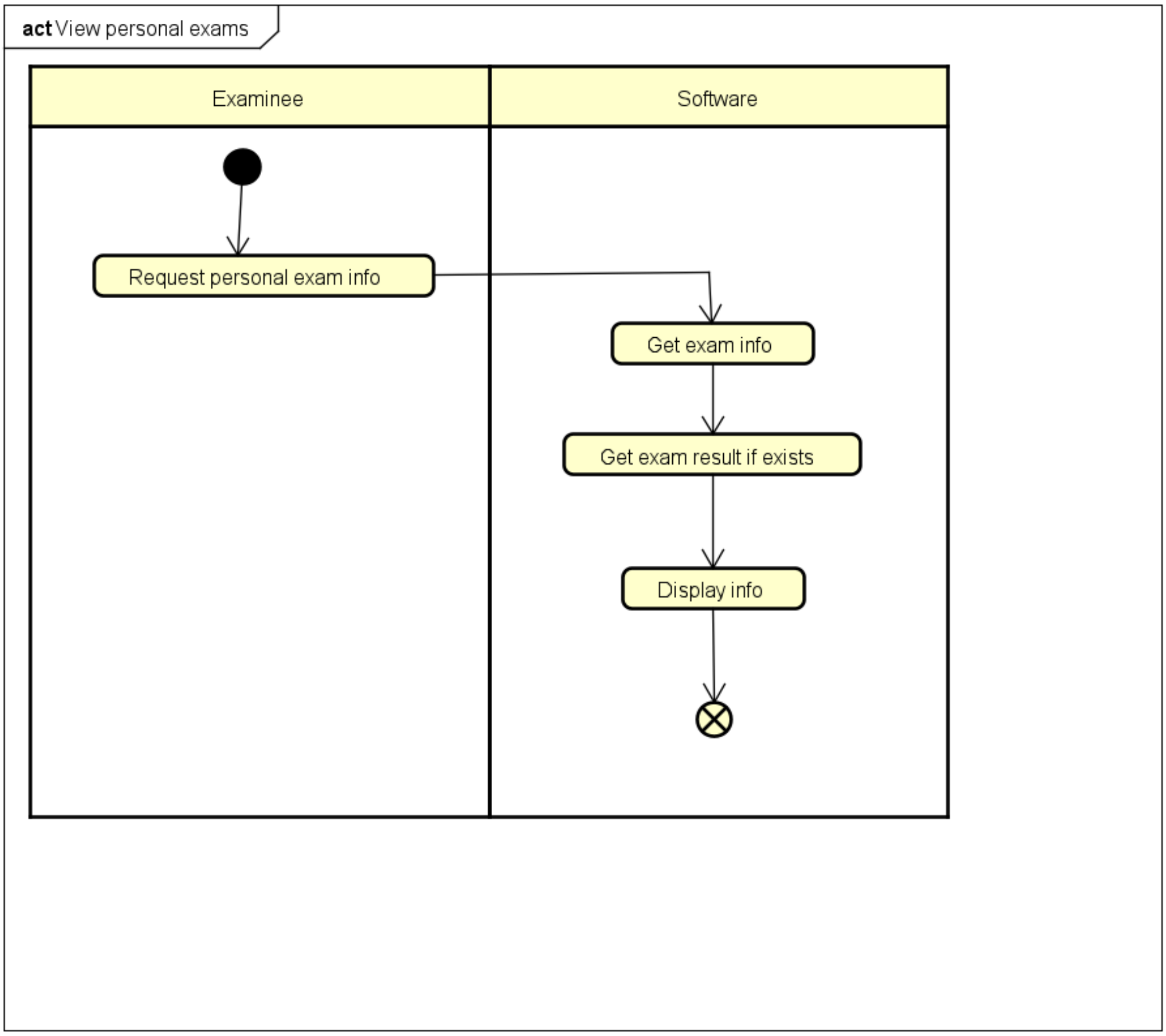

Figure11. View Personal Exam Activity Diagram

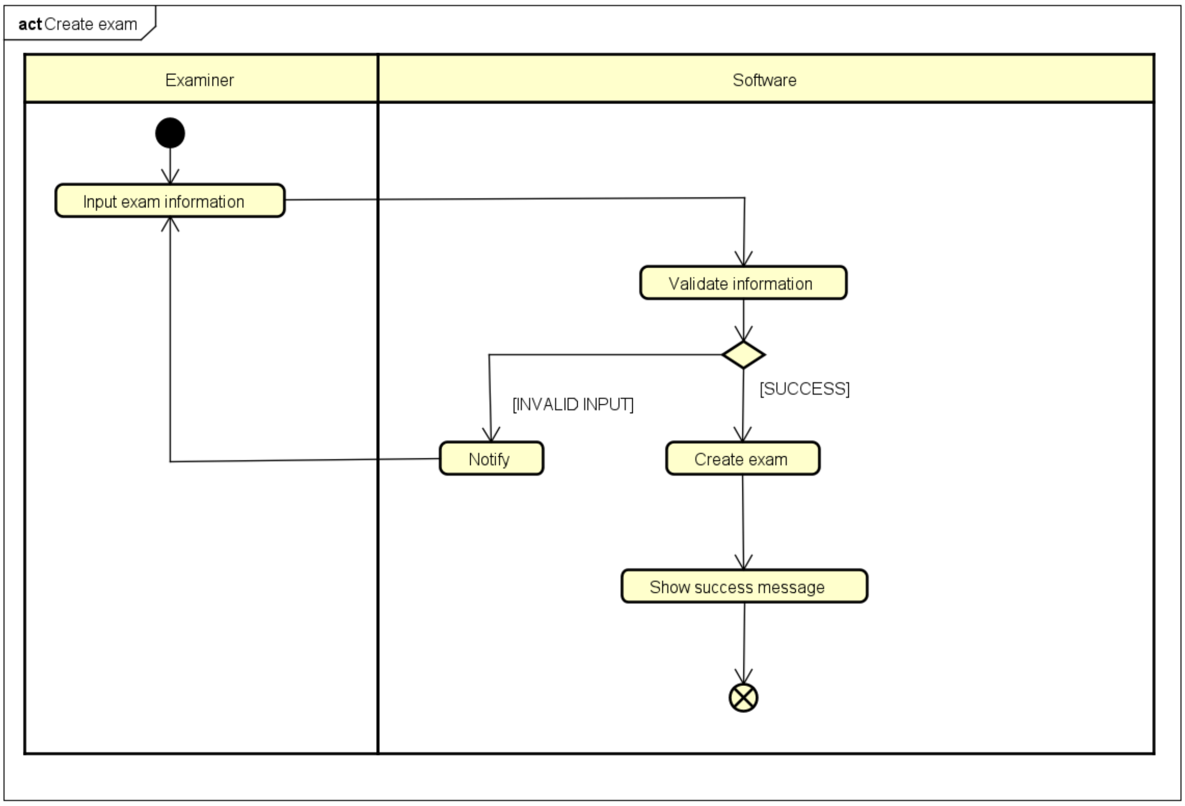

Figure12. Create Exam Activity Diagram

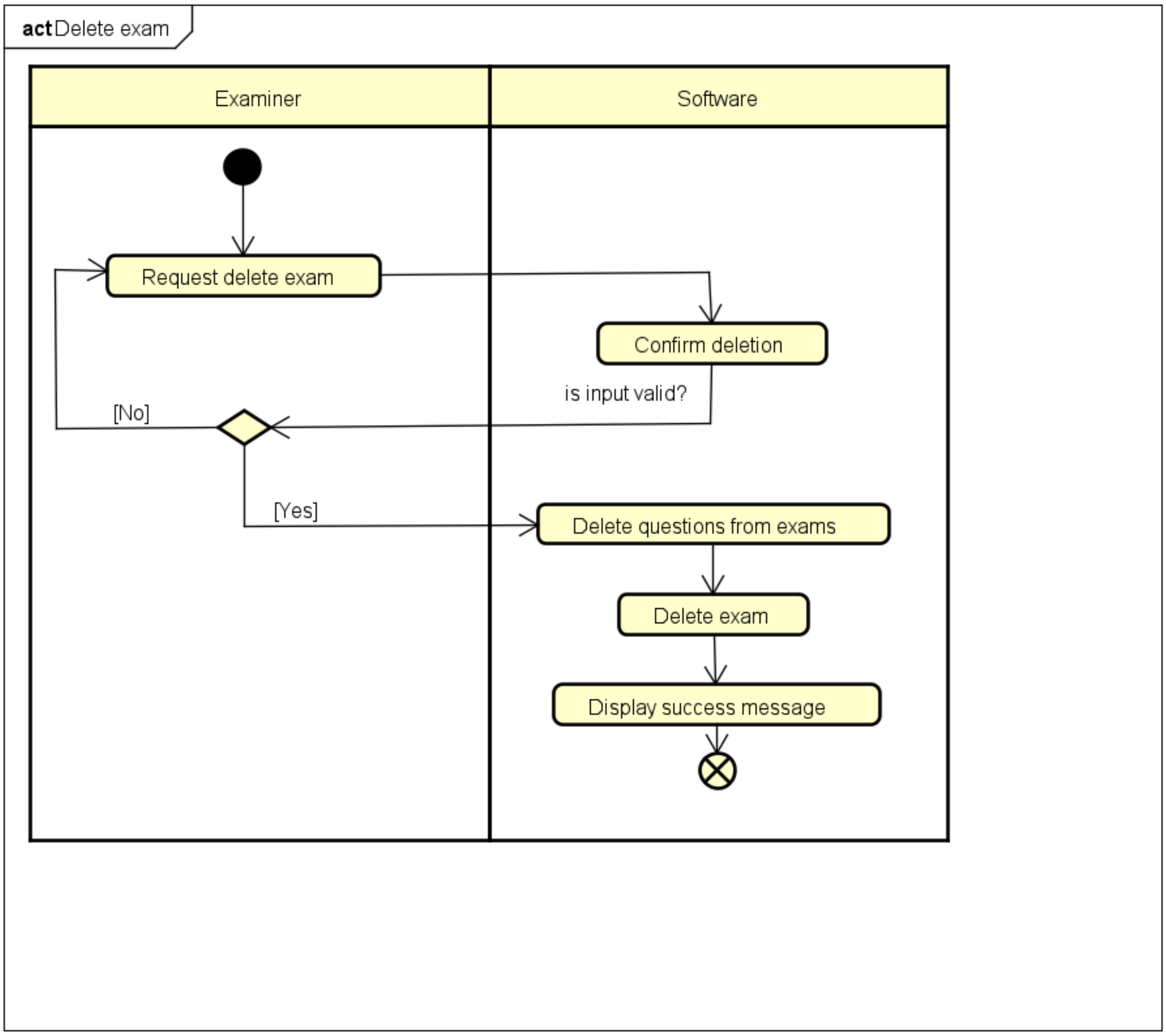

Figure13. Delete Exam Activity Diagram

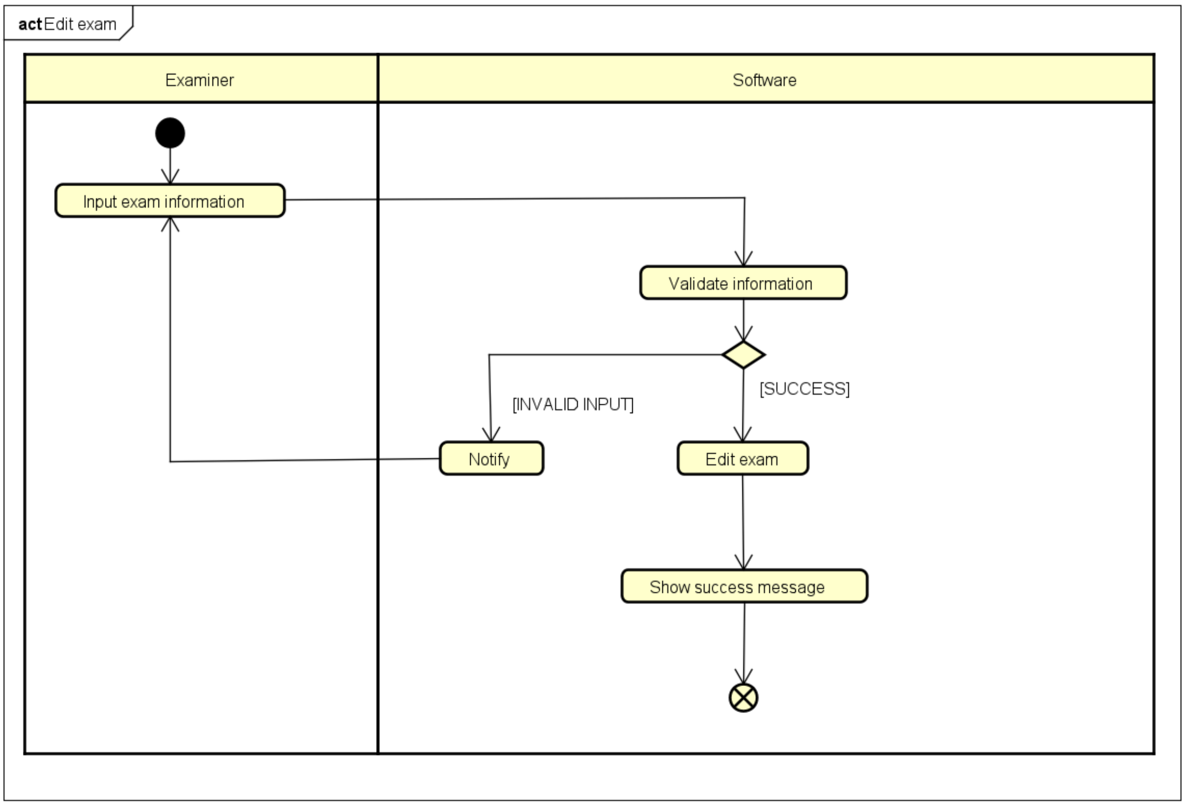

Figure14. Edit Exam Activity Diagram

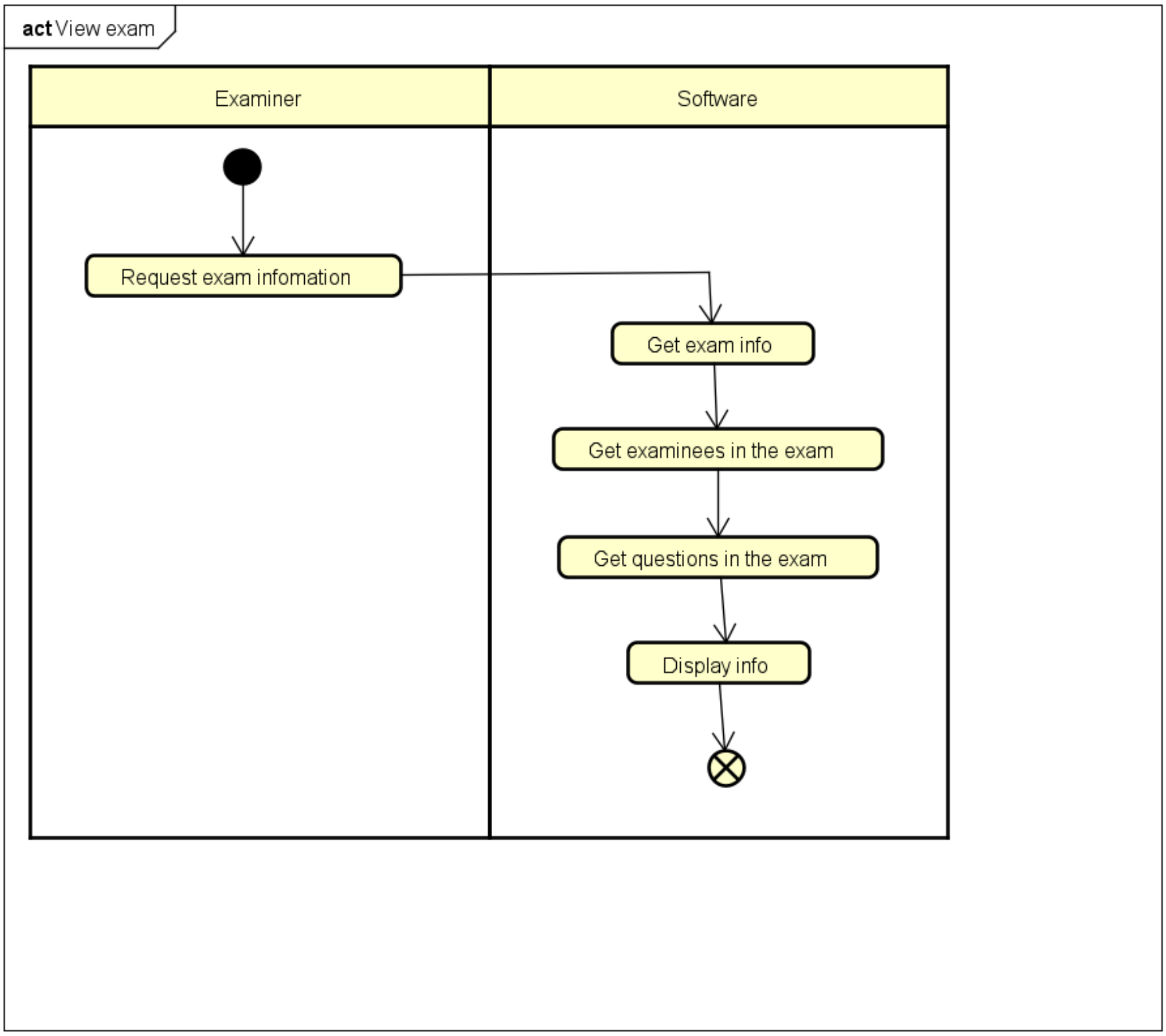

Figure15. View Exam Activity Diagram

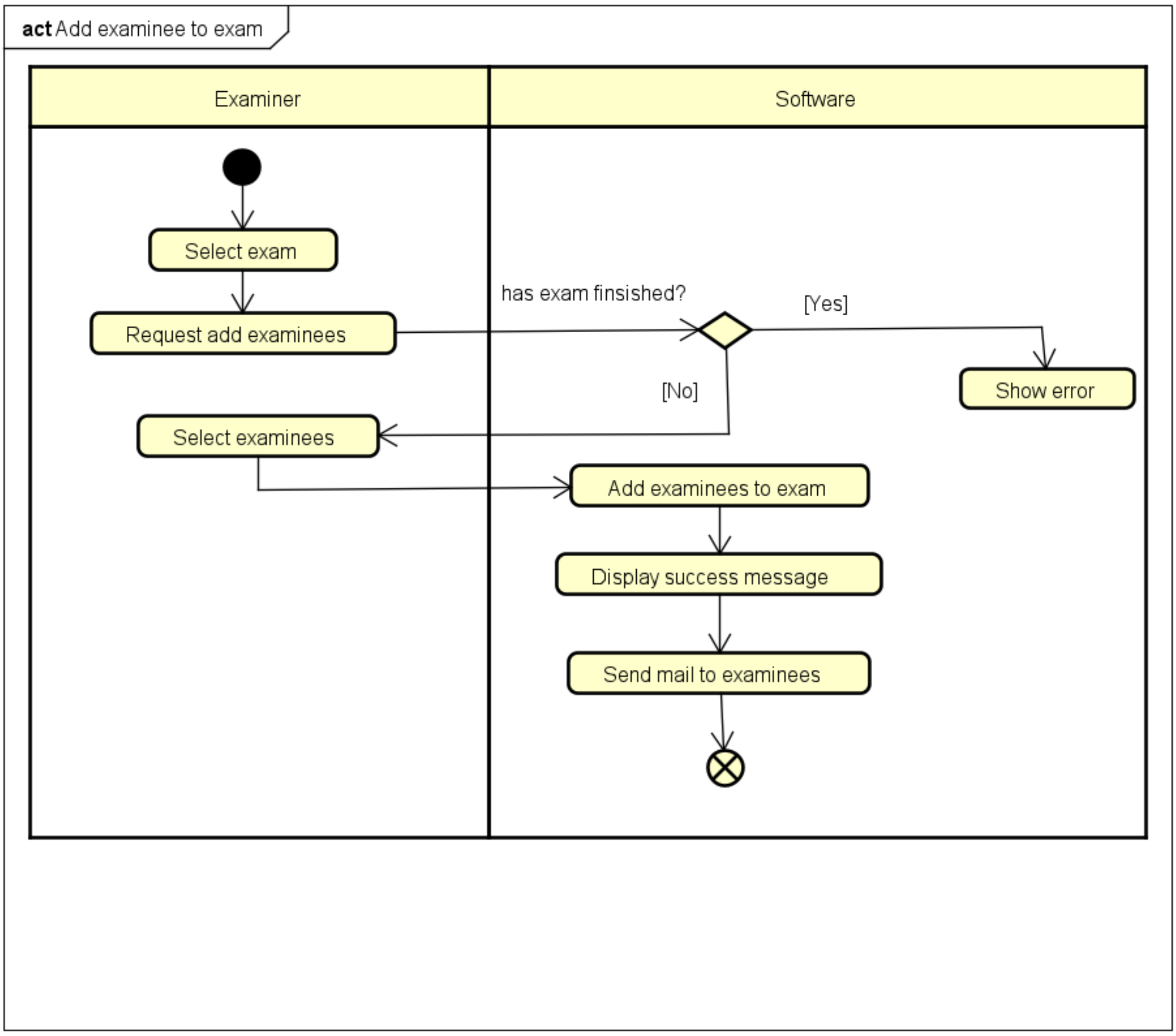

Figure16. Add Examinee to Exam Activity Diagram

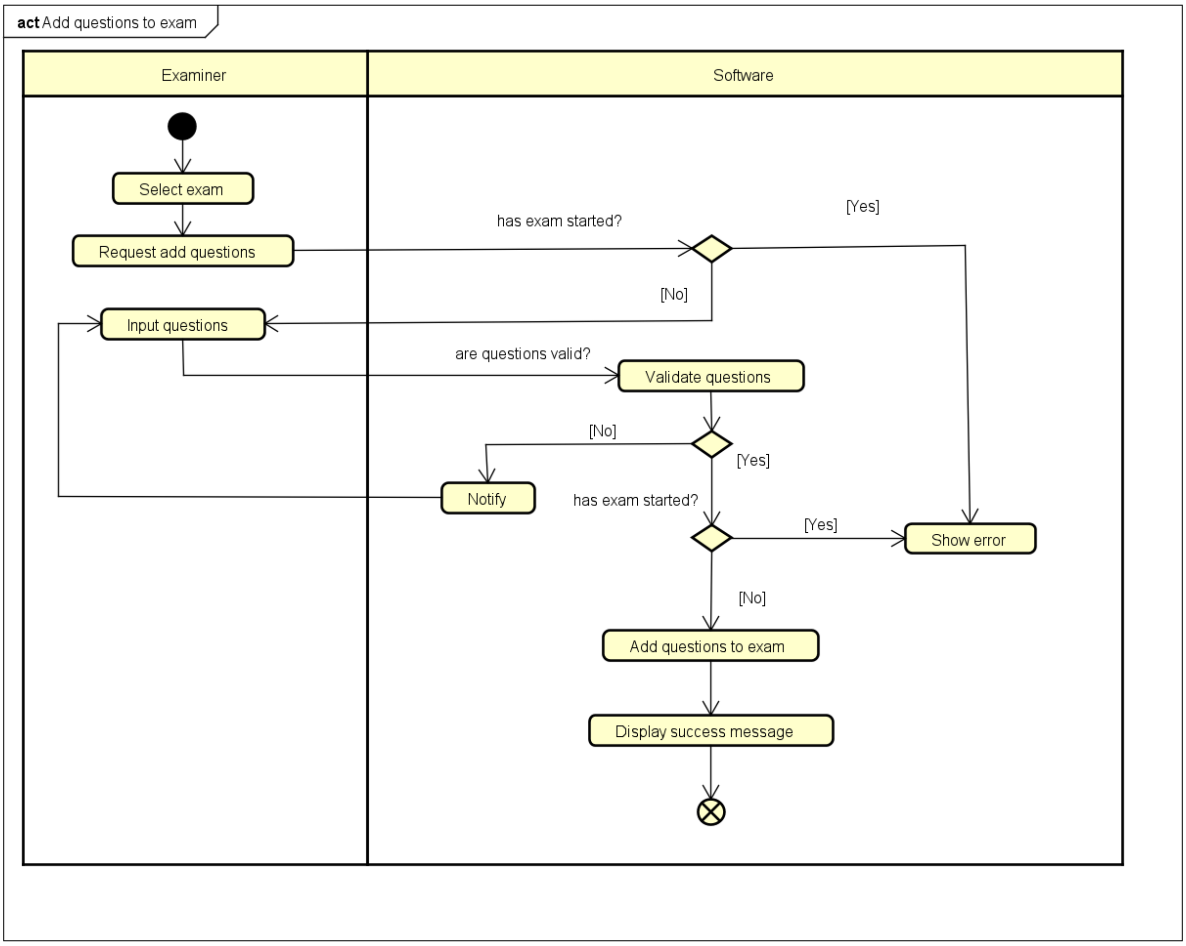

Figure17. Add Questions to Exam Activity Diagram

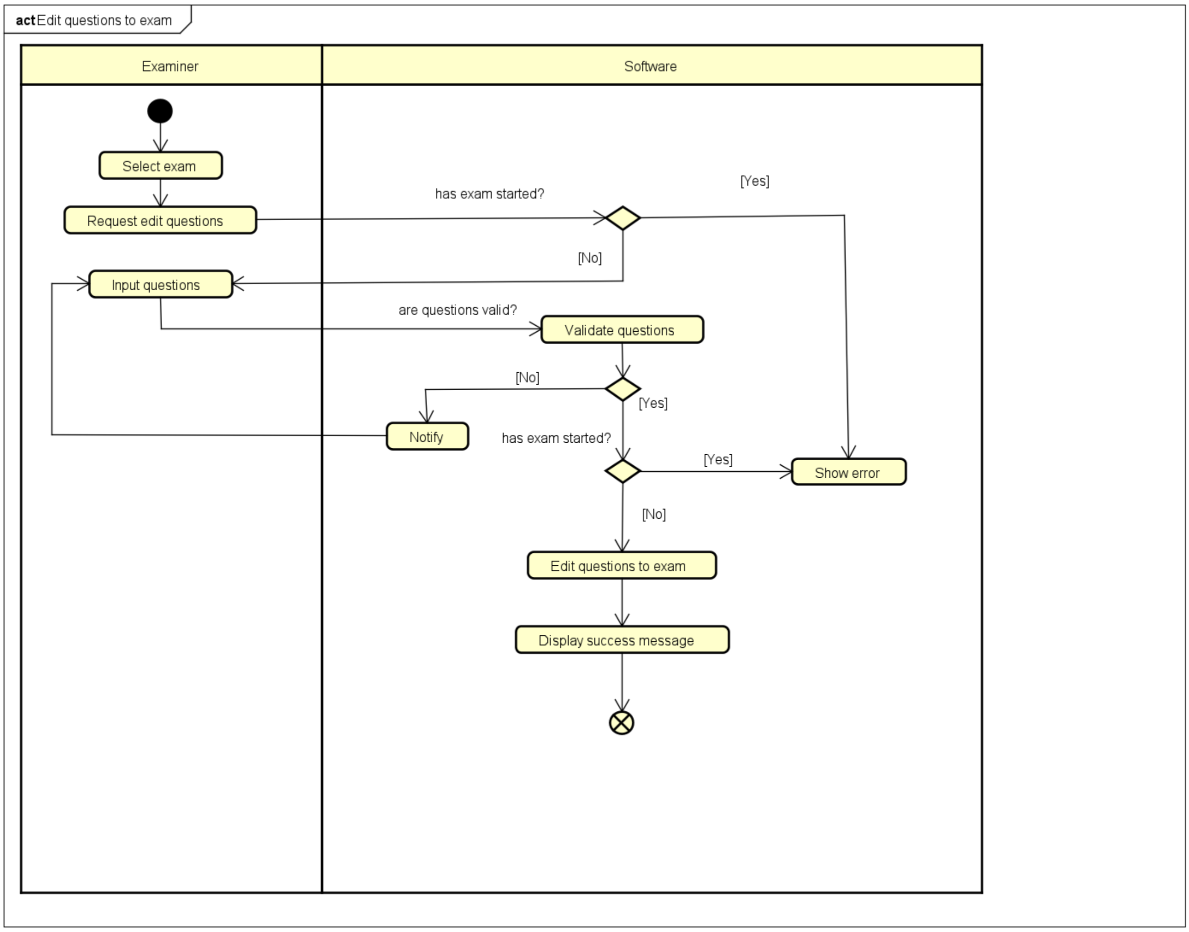

Figure18. Edit Exam Question Activity Diagram

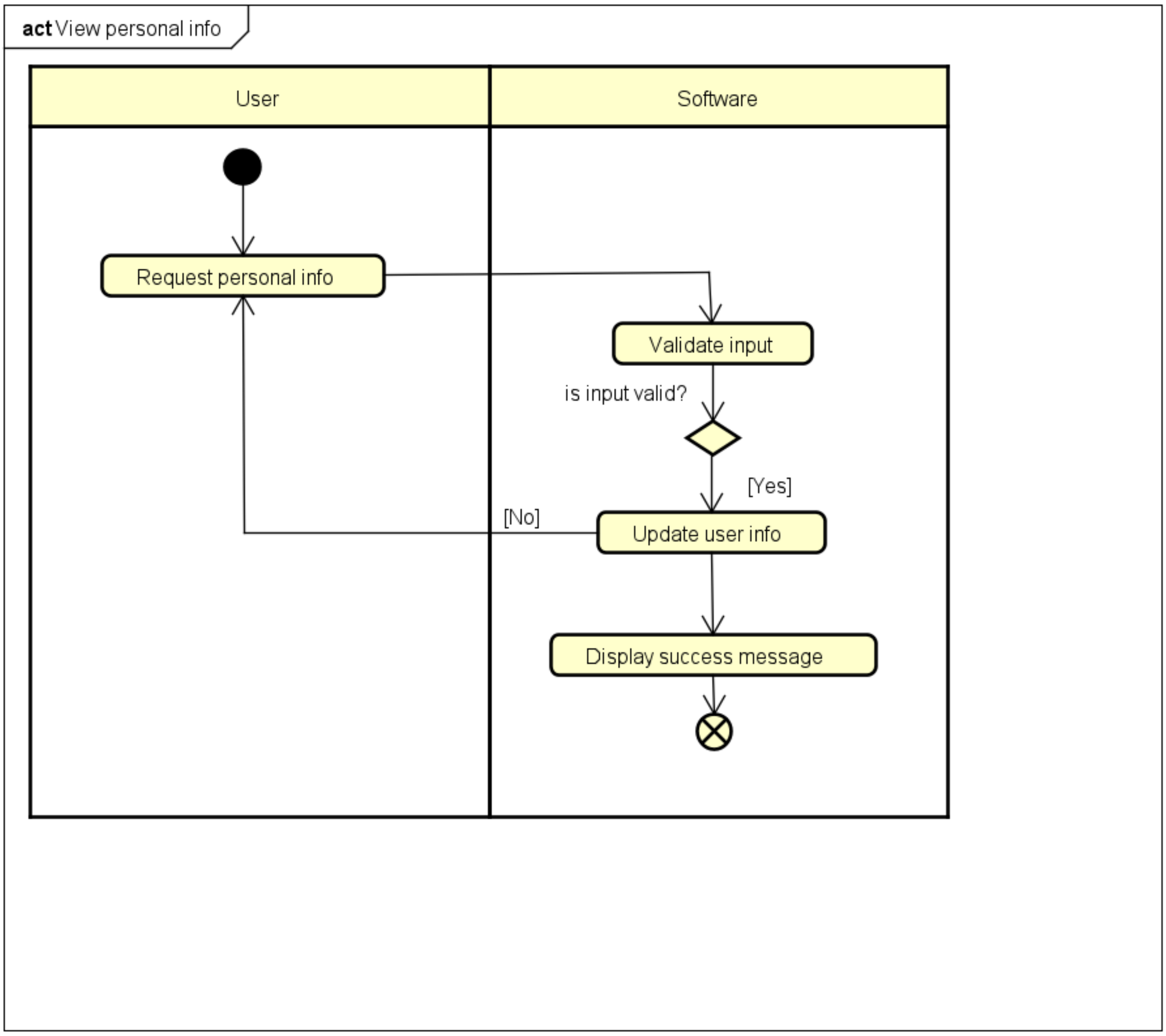

Figure19. View Personal Info Activity Diagram

2.1 Product perspective

2.1.1 System interfaces

The system exposes several APIs to have client side interact with:

- Login, CRUD accounts to interact with accounts
- CRUD exam, add/modify participants, questions, answers to interact with the exam
- Load questions, save answers, save log to record the examinee’s exam progrogress
- View log, view result to see data of the past exams
- General webview interface to load the website.

2.1.2 User interfaces

a, Logical characteristic:

- Support for desktop and tablet size, and optionally mobile devices.
- Window layout: 1080 in width for desktop, 720 for tablet.
- Content:
    + For login page: username and password input
    + For CRUD account: name, email, password, date of birth, phone number and other info input based on the type of account
    + For CRUD exam: exam name, exam scoring rules, exam start time and duration, add and remove questions, participants input.
    + Exam participation page: display of questions and answers, with ability to choose one or many answers, can navigate to next question or submit the exam
    + View log page: view existing logs, with ability to add filters based on time and account.
    + Grading page: display of questions, choices and answers of the participant, with optional input to directly give points.

b, Target
The exam interface should be simple and intuitive for examinees, while interface for admins and examiners might take some practice. We aim that an examiner can be proficient in using this system after a week, and a few days for the administrator.

2.1.3 Hardware interfaces

As with other web-based software, this program exposes port 80 to serve the web. This program is intended to be run on a Unix-based server or through containerization with Docker, and on the client side are browsers such as Chrome or Firefox. This program will only provide UI support, with some admin-level commands that might be run through local shell scripts.

2.1.4 Software interfaces

The list of external softwares used:

- Postgres: a database
    + Version: 14.0
    + Source: https://github.com/docker-library/postgres/tree/c6329e3bf217ca53fbb78d27d756f95498cb143f/14
    + Interface: https://www.postgresql.org/docs/14/external-interfaces.html

2.1.5 Communications interfaces

Communication between the frontend and the backend is through HTTP requests.

2.1.6 Memory

On the server, there is not a hard constraint, but generally storage should be saved per semester. Caching content at a given time will primarily be exam questions and answers.

On the client side, data of the account can be saved locally for each session. During the exam, questions will be sent one at a time, and therefore do not have any considerable effect on the memory.

2.1.7 Operations

a, Operations list

- For examinee: modify information, do exam and view results
- For examiner: create exam, modify exam, participants to the exam, questions, answers, scores, view exam log.
- Admin: manage accounts, view system logs.

b, Backup and recovery

- Backup on a 24 hours basic through cron job
- Recovery has to be done manually by technicians.

2.1.8 Site adaptation requirements

The software needs to be scalable with the number of concurrent examinees interacting with the system at the same time, which is approximately between 900 and 30000.

2.2 Product functions

The product's functions are as follow

- Manage User database
- Manage Exam database
- Automatic exam holding
- Automatic exam scoring

The product provides the following functions to users:

- Admin
  - CRUD Accounts
  - View exam log
- Examiner:
  - CRUD exam
  - Manually give score
  - View exam report
  - View progress of students in an on-going exam
- Examinee:
  - Take exam
  - View personal exam
  - Search for exams
- All types of users can:
  - View Personal info
  - Edit personal info

2.3 User characteristics

The characteristics for intended users are as follow:

- Education Level:

    - The Examiner should have a Teaching Certificate of any subject
- Experience:
    - No experience is required
- Technical expertise:
    - The Examiner should have basic office computer understanding
    - The Examiner should be able to use Latex (For Mathematics Exams)
    - The Examinee should have basic office computer understanding
    - The Admin should have basic office computer understanding

2.3 Constraints

The known constraints are as follow:

- regulatory policy: This product should be available for a group of students, teachers and administrators of a school. The school should therefore have full control of the functionality and configuration of the program.
- hardware limitation: should be light and fast on the user device to not affect their examination.
- reliability requirement: software should be resilient to bursts of requests. Due to the non-deterministic nature of network transmission, each step in an examinee's exam progress should be recorded on the server, and can be used to restore the last session if the examinee is disconnected from the exam portal.
- Criticality of the application: there should not be any unrecoverable error when an exam is happening, because it will affect the validity of the exam, and the fairness for each participant.
- Security consideration: because the server will store exam logs and results of the examinee, it is important to secure access to the examiner and admin level account. Additional counter-measure should also be installed to prevent attacks against the integrity of data, and availability of the system, especially during an exam’s time frame.

2.4 Assumptions and dependencies

The assumptions for the product are as follow:

- All users should be equipped with personal computers that can access the Internet
    - The personal computers are installed with Windows/Linux/MacOS, and have web browsers with GUI.

2.5 Apportioning of requirements

Due to time, human resource and monetary constraint, this project will need to delay some features which are required of a full-fledged online exam software, notably:

- integration with existing databases: usually a school/university will have a database of students and their subjects, but because we are restricted to our mock data only, we can not implement this feature truthfully.
- anti-cheat system: using this web-based examination solution is not completely fool-proof because the browser can be quite unreliable in detecting examiners' gestures and behaviours, therefore restrictions from the software using browser’s API

interface is not guaranteed to work without extensive implementation and testing. We would prefer dedicating our limited resources to fundamental features before working on this task.

- Some security features such as HTTPS, anti-DDOS might not be implemented because such a task requires external dependencies, and while not really affecting the development process. We will consider this task when deploying this software.