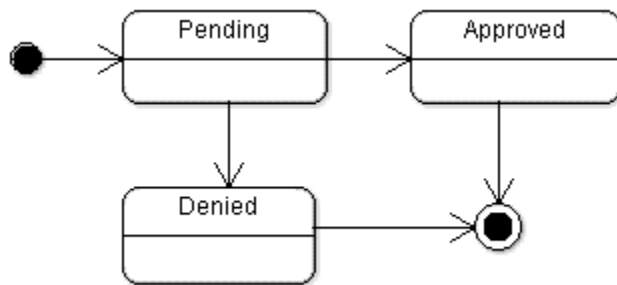# Expense Reimbursement System

Executive Summary

The Expense Reimbursement System (ERS) will manage the process of reimbursing employees for expenses incurred while on company time. All employees in the company can login and submit requests for reimbursement and view their past tickets and pending requests. Finance managers can log in and view all reimbursement requests and past history for all employees in the company. Finance managers are authorized to approve and deny requests for expense reimbursement.
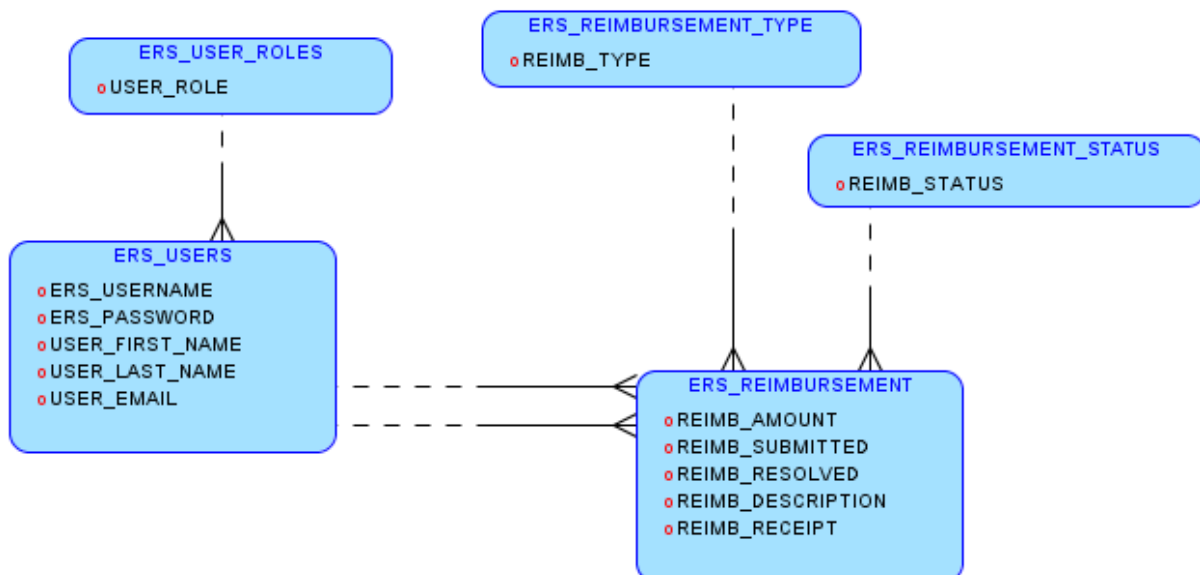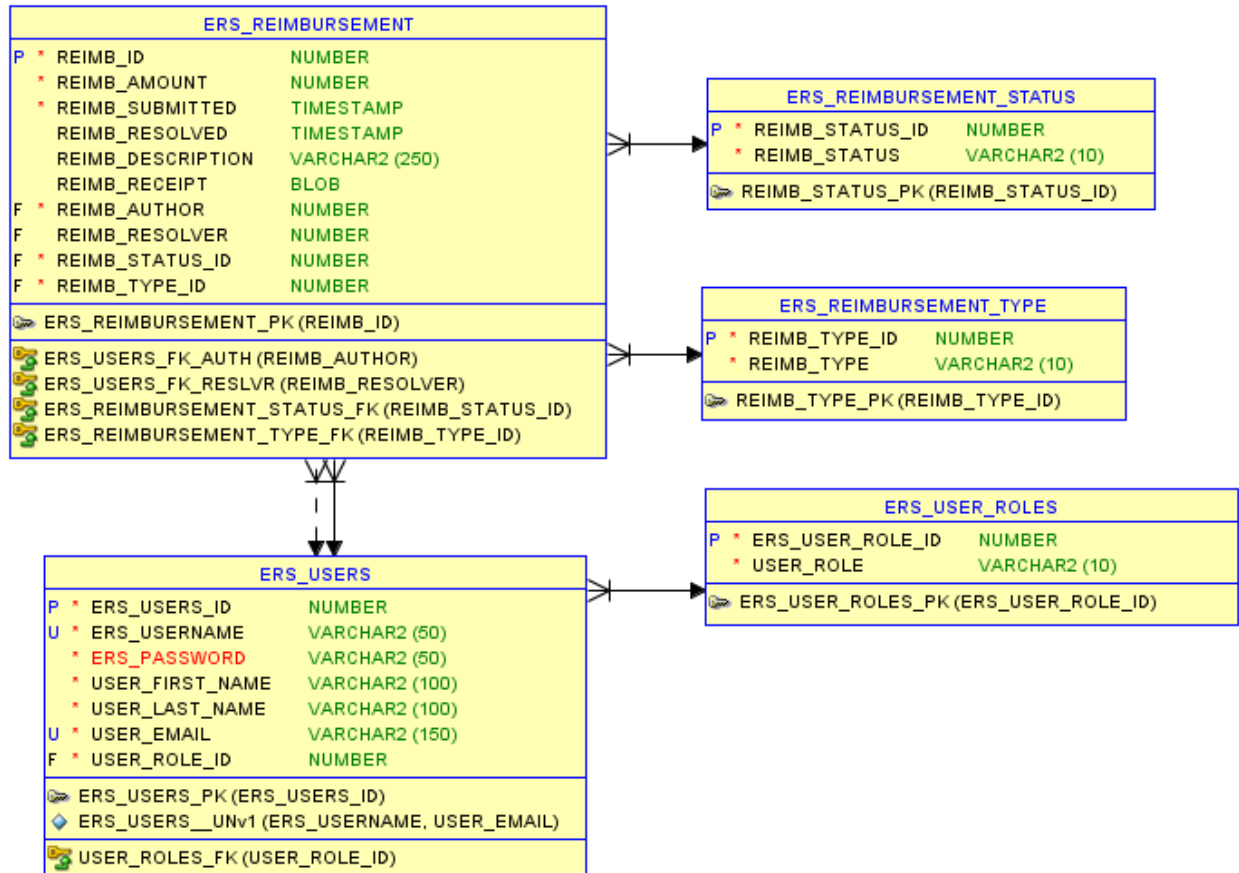
State-chart Diagram (Reimbursement Statuses)

Reimbursement Types

Employees must select the type of reimbursement as: <u>LODGING</u>, <u>TRAVEL</u>, <u>FOOD</u>, or <u>OTHER</u>.
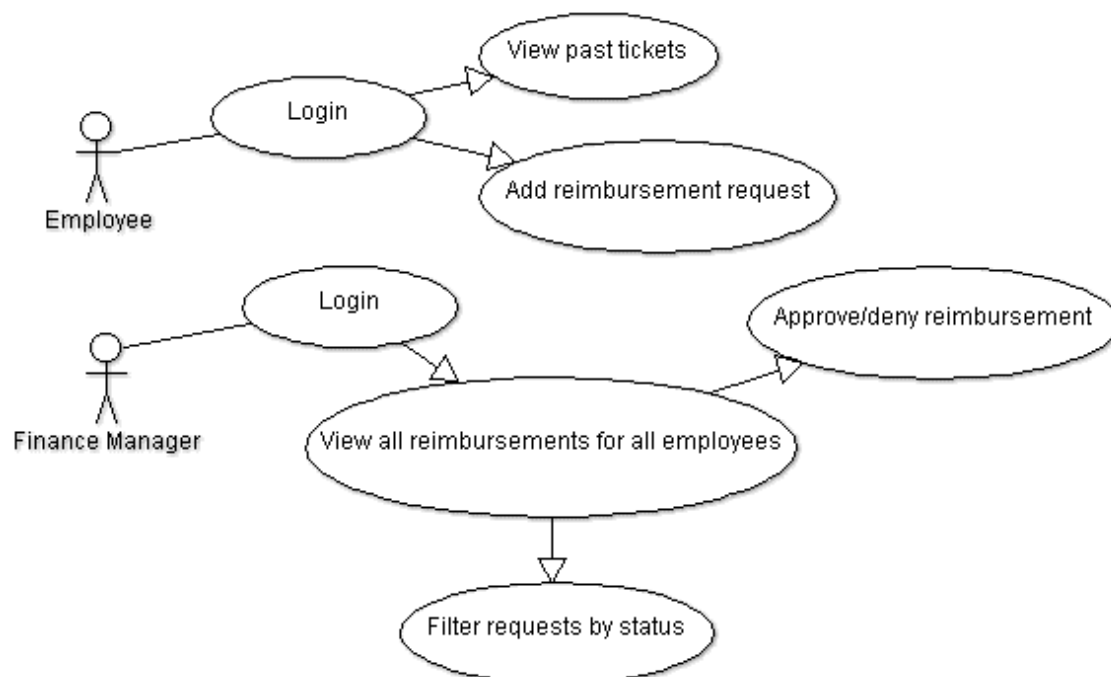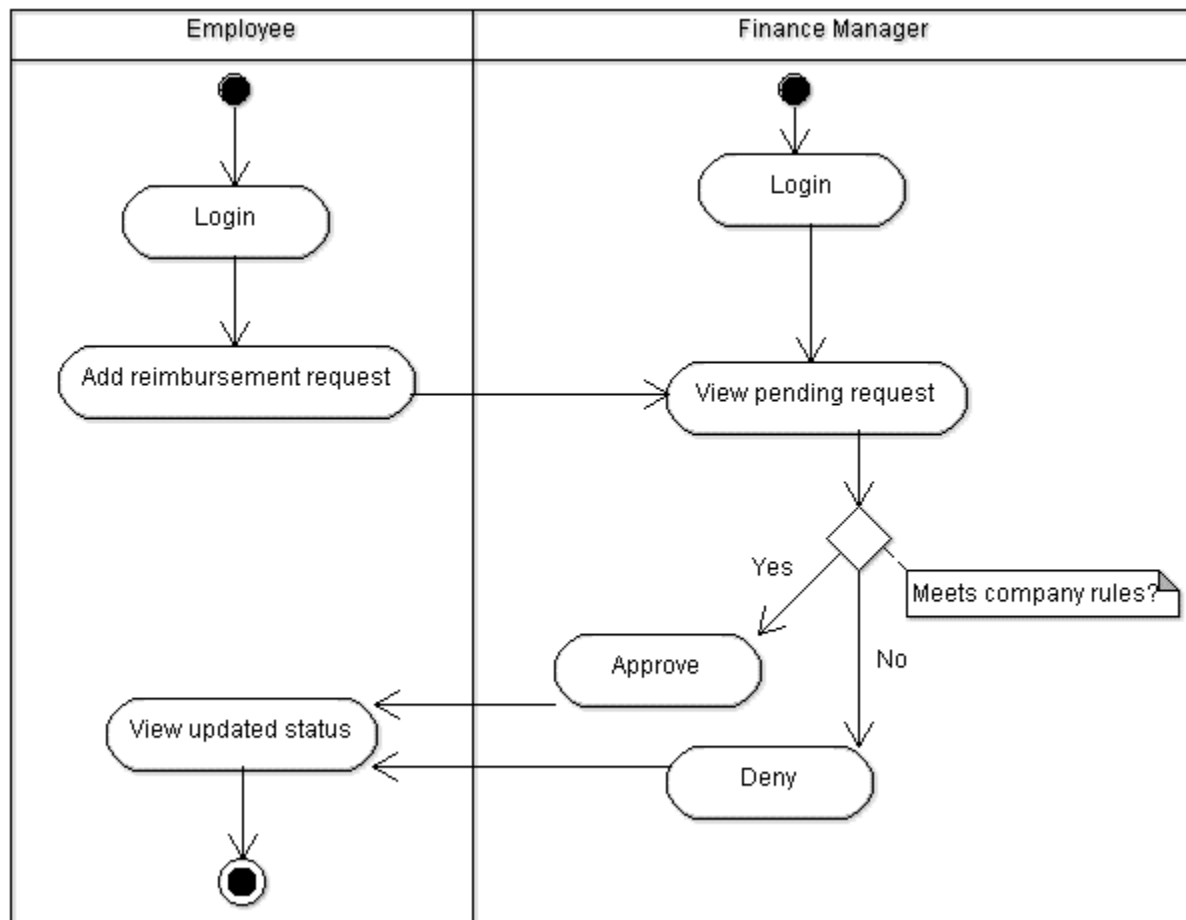
Logical Model

Physical Model

ERS_REIMBURSEMENT

P	* REIMB_ID	NUMBER
	* REIMB_AMOUNT	NUMBER
	* REIMB_SUBMITTED	TIMESTAMP
	REIMB_RESOLVED	TIMESTAMP
	REIMB_DESCRIPTION	VARCHAR2 (250)
	REIMB_RECEIPT	BLOB
F	* REIMB_AUTHOR	NUMBER
F	REIMB_RESOLVER	NUMBER
F	* REIMB_STATUS_ID	NUMBER
F	* REIMB_TYPE_ID	NUMBER

ERS_REIMBURSEMENT_PK (REIMB_ID)

ERS_USERS_FK_AUTH (REIMB_AUTHOR)
ERS_USERS_FK_RESLVR (REIMB_RESOLVER)
ERS_REIMBURSEMENT_STATUS_FK (REIMB_STATUS_ID)
ERS_REIMBURSEMENT_TYPE_FK (REIMB_TYPE_ID)

ERS_REIMBURSEMENT_STATUS

P	* REIMB_STATUS_ID	NUMBER
	* REIMB_STATUS	VARCHAR2 (10)

REIMB_STATUS_PK (REIMB_STATUS_ID)

ERS_REIMBURSEMENT_TYPE

P	* REIMB_TYPE_ID	NUMBER
	* REIMB_TYPE	VARCHAR2 (10)

REIMB_TYPE_PK (REIMB_TYPE_ID)

ERS_USER_ROLES

P	* ERS_USER_ROLE_ID	NUMBER
	* USER_ROLE	VARCHAR2 (10)

ERS_USER_ROLES_PK (ERS_USER_ROLE_ID)

ERS_USERS

P	* ERS_USERS_ID	NUMBER
U	* ERS_USERNAME	VARCHAR2 (50)
	* ERS_PASSWORD	VARCHAR2 (50)
	* USER_FIRST_NAME	VARCHAR2 (100)
	* USER_LAST_NAME	VARCHAR2 (100)
U	* USER_EMAIL	VARCHAR2 (150)
F	* USER_ROLE_ID	NUMBER

ERS_USERS_PK (ERS_USERS_ID)
ERS_USERS__UNv1 (ERS_USERNAME, USER_EMAIL)

USER_ROLES_FK (USER_ROLE_ID)

Use Case Diagram

Employee — Login — View past tickets
Login — Add reimbursement request

Finance Manager — Login — View all reimbursements for all employees — Approve/deny reimbursement

View all reimbursements for all employees — Filter requests by status

Activity Diagram

Technical Requirements

The back-end system should use a NodeJS server to communicate with a PostgreSQL database. You may use pg, knux, or any appropriate ORM technology for communication. The NodeJS application must expose a RESTful API that meets the specifications for level 2 on the Richardson Maturity Model and follows general resource naming conventions. You must build a client-side application which consumes this API and offers all functionality described in the activity and use-case diagrams. The client-side application may be built using HTML/CSS/JS or using React. Additionally, you should include some client-side design library such as Bootstrap or Material-UI. User Experience will be a consideration for evaluation of this project.

Testing Goals
- >80% test coverage for server
- >50% test coverage clientside

Process Recommendations
- Commit AND push code daily
- Communicate blocking problems that impede progress promptly

Optional stretch goals:

- Supervisor or auditor roles which can audit decisions made by finance managers.
  - Allow users with denied requests to appeal decisions to auditors.
- Enforce special requirements for very expensive reimbursements.
- More detailed/fine-tuned search functionality.
- Other appropriate features that suit the application