**Project Design**

**CMSC 496 – 6380**

**Group 8**

**Robert Hunter Robinson**

**Jered Russell**

**Keith Tulloch**

**Program Name: Forget Me Not**

**Date of Revision: 10/11/2018**

**Revision Number: 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date** | **Description** | **Name** |
| 1 | 9/15/2018 | Add startup, shutdown, and normal scenarios | Keith |
| 2 | 9/16/2018 | Added Java and SQL Pseudocode | Hunter |
| 3 | 9/16/2018 | Added error-handling trace diagram and unresolved risk and risk mitigation | Jered |
| 4 | 9/16/2018 | Added 2nd option to risk #1 mitigation | Hunter |
| 5 | 10/07/2018 | Changed verbiage of Scenarios 1 and 2 to match other documents. Deleted Error Handling Scenario 3 that is no longer required. | Jered |
| 6 | 10/11/2018 | Added note for references to SQL database with .txt document due to inability to secure an online SQL server for program. | Hunter |

**Event-trace diagrams:**

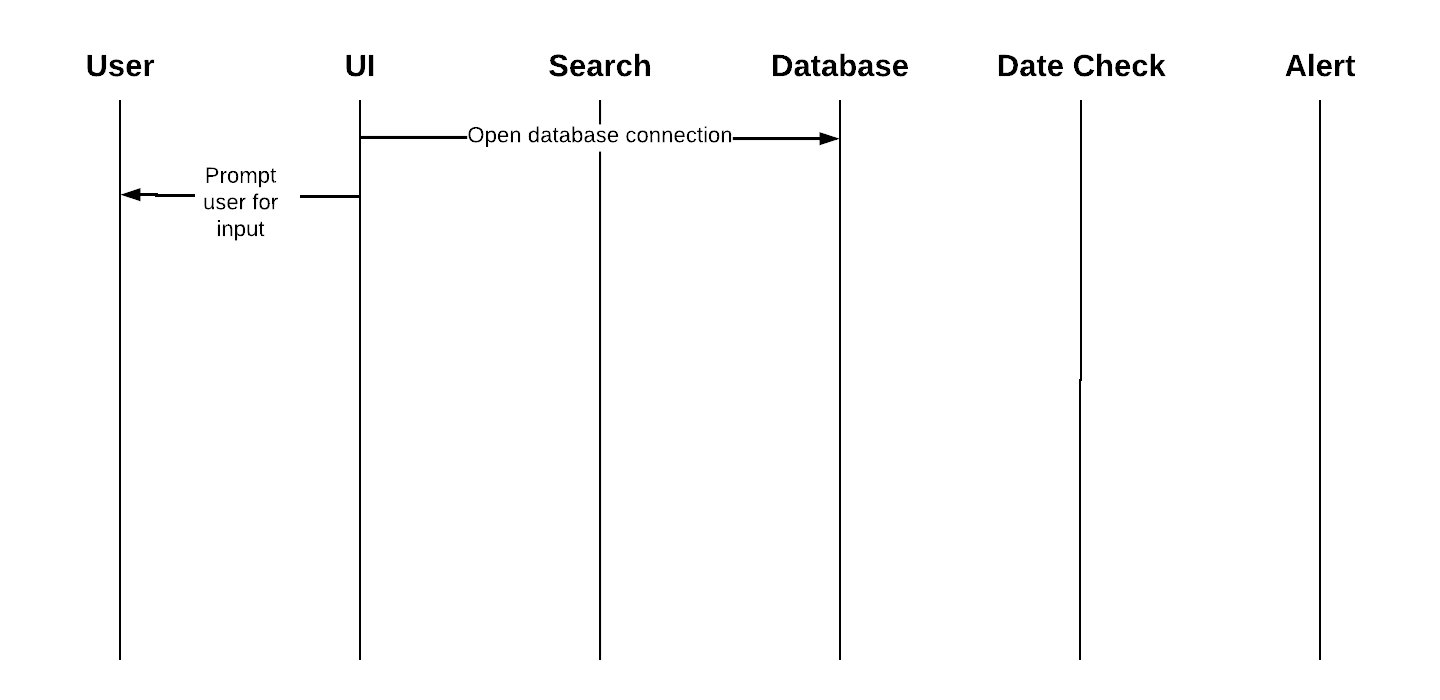
Note: all references to databases are being simulated by a connection to a .txt document.

**Startup scenario:**

Description: open connection to database and prompt user for input

Precondition: system has adequate resources to run

Post-condition: connected to database; application awaits user input or change to current date

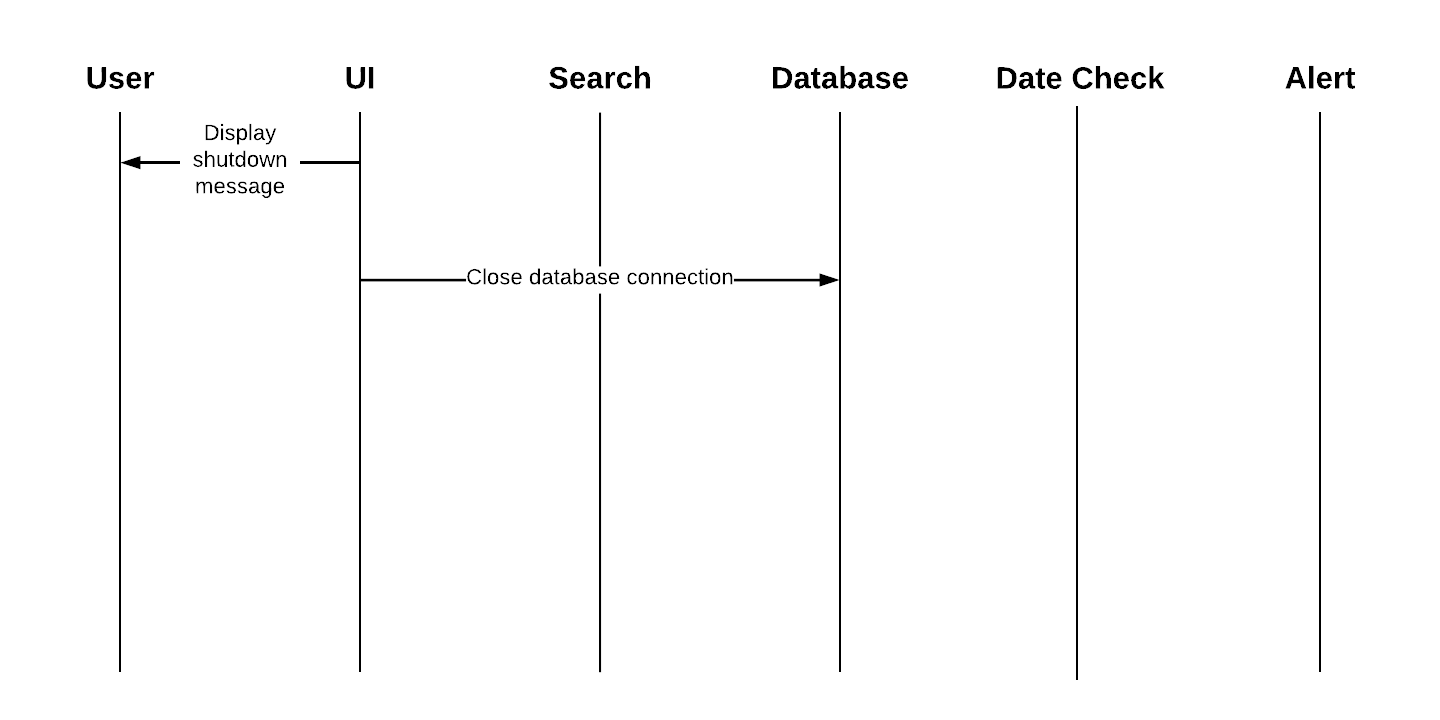


**Shutdown scenario:**

Description: display shutting down message; close connection to database

Precondition: application is running

Post-condition: database connection stopped; application closed

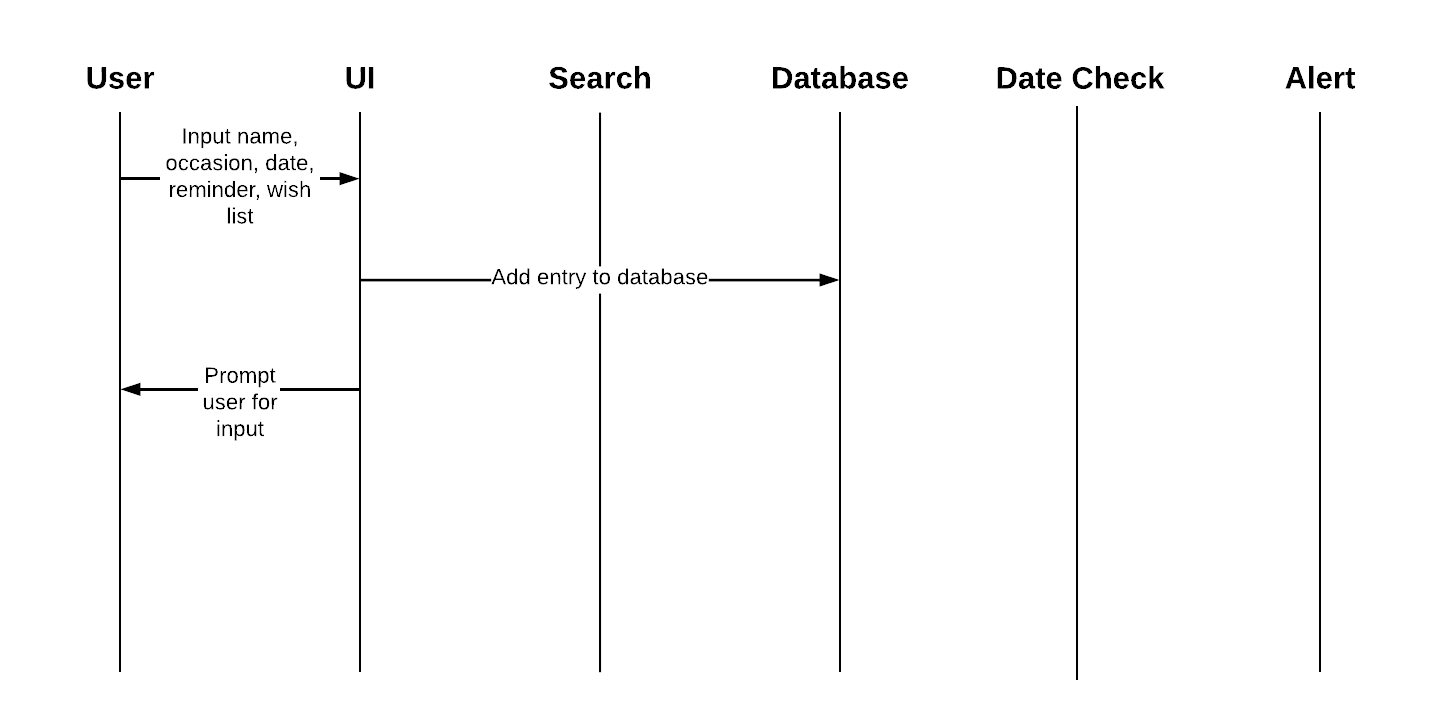


**Scenario 1:**

Description: the user adds a complete special occasion entry to be stored in database

Precondition: application is running; connection to database

Post-condition: entry is added to database

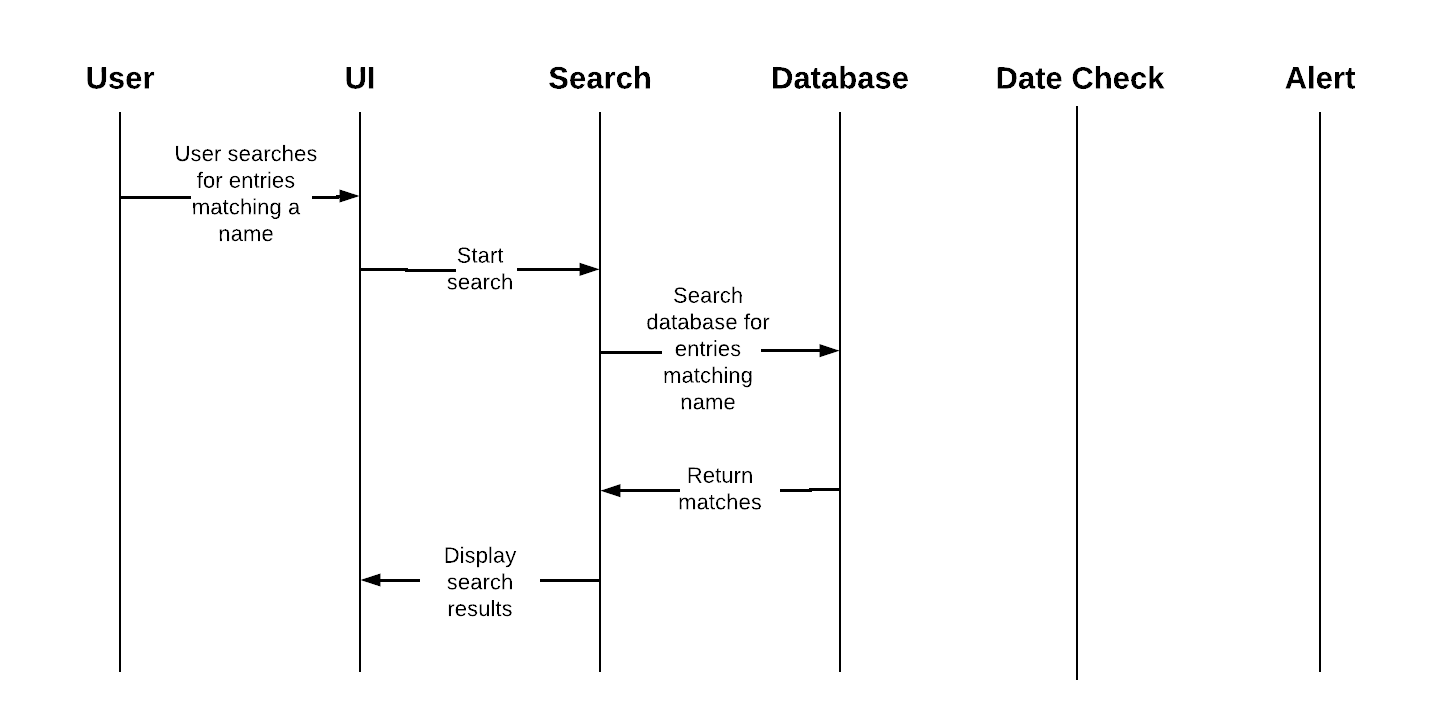


**Scenario 2:**

Description: the user searches for a special occasion previously added to database by name

Precondition: application is running; connection to database

Post-condition: entries matching the name are displayed to user

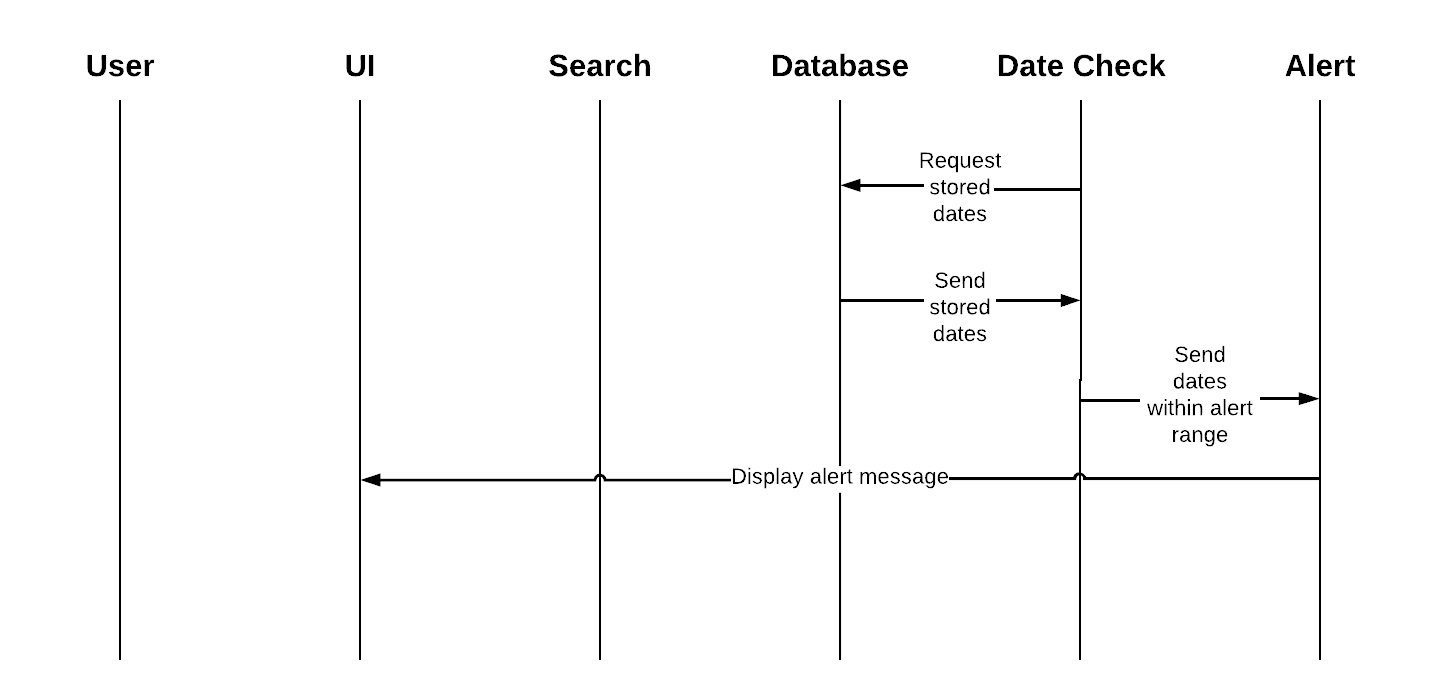
****

**Scenario 3:**

Description: system running on or near a day that has a saved occasion reminder alert set

Precondition: application is running; connection to database

Post-condition: user notified of any upcoming occasions and/or events on current date

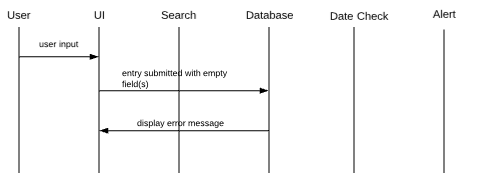


**Error-Handling Scenario 1:**

Description: user submits entry with one or more empty fields

Precondition: application is running

Post-condition: user is notified by an error message to complete all fields

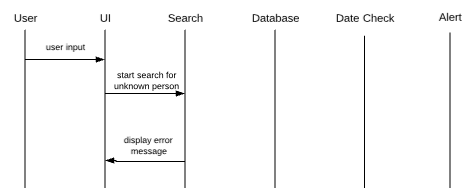
****

**Error-Handling Scenario 2:**

Description: user searches for a person who is not stored in database

Precondition: application is running; connection to database

Post-condition: user is notified with error message that person does not exist in database



**Pseudocode:**

Variable, method, class names, and wish list size subject to change.

**Java:**

GUI DETIALS NOT INCLUDED.

**Main Class: (User Interface Subsystem)**

Class ForgetMeNot

GUI

String name

String occasion

Date eventDate = new Date(int year, int month, int date);

Date reminderDate = new Date(int year, int month, int date);

String[4] wishlist

Main method

**SQL Processes: (Handles input from User interface, Alert, and Search Subsystems)**

Class ForgetNotProcess

Method to convert input variables into an SQL statement as a string and pass to SQL database (User Interface to Database Subsystems)

Method to pass SQL statements to SQL database to handle search (Search Subsystem)

Method to receive date alerts from SQL and pass alerts to User Interface (Alert Subsystem)

**SQL: (Database and Date Check Subsystems)**

**Table Creation:**

CREATE TABLE FORGETNOT (

date\_ID int NOT NULL AUTO\_INCREMENT,

name VARCHAR2(30) NOT NULL,

occasion VARCHAR2(30) NOT NULL,

event\_date DATE NOT NULL,

reminder\_date DATE NOT NULL,

item1 VARCHAR2(20),

item2 VARCHAR2(20),

item3 VARCHAR2(20),

item4 VARCHAR2(20),

CONSTRAINT forgetnot\_pk PRIMARY KEY (date\_ID)

);

**Upon receiving input from Java:**

INSERT INTO FORGETNOT (name, occasion, event\_date, reminder\_date, item1, item2, item3, item4)

VALUES (‘name’, ‘occasion’,

TO\_DATE(‘event\_date’, ‘MM/DD/YYYY’), TO\_DATE(‘reminder\_date’,’MM/DD/YYYY’),

‘item1’, ‘item2’, ‘item3’, ‘item4’);

**Java Date query:**

**Days Left:**

SELECT reminder\_date FROM FORGETNOT f WHERE f.occasion=x.occasion;

**Alert/Search Retrieval:**

SELECT \* FROM FORGETNOT WHERE DATEDIFF(day, reminder\_date,

GETDATE()) = 0;

**Unresolved Risks and Risk Mitigation:**

**Risk #1:** Unauthorized user on local machine

**Possible Mitigation #1:**  Create a logon for access to Forget Me Not

**Possible Mitigation #2:**  Serialize the program so that only proper serials will work