

Gebze Technical University  
Computer Engineering

CSE 222  
2017 Spring

HOMEWORK HW01 REPORT

Burak AKTEN  
141044045

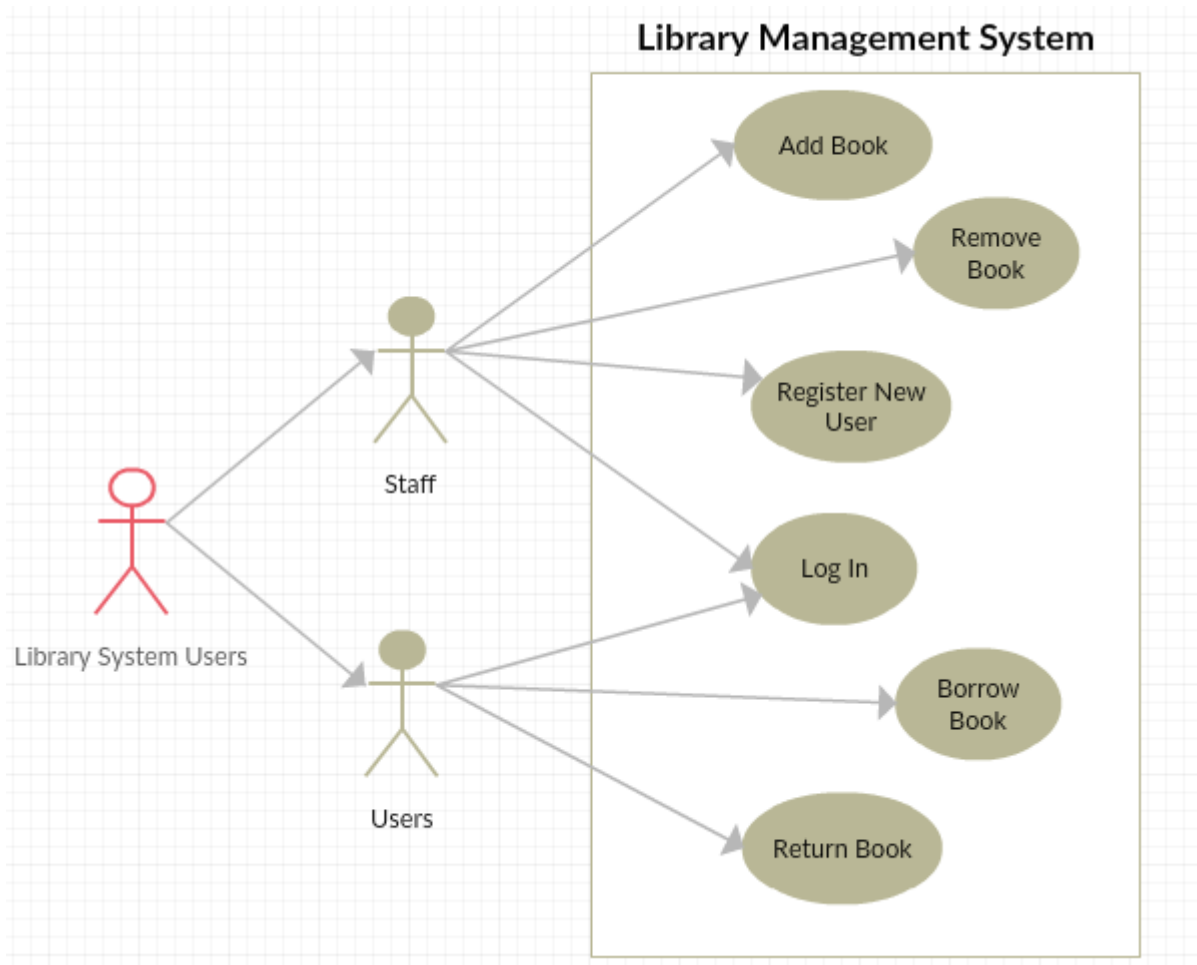
Course Assistant: Nur Banu Albayrak

## 1. System Requirements

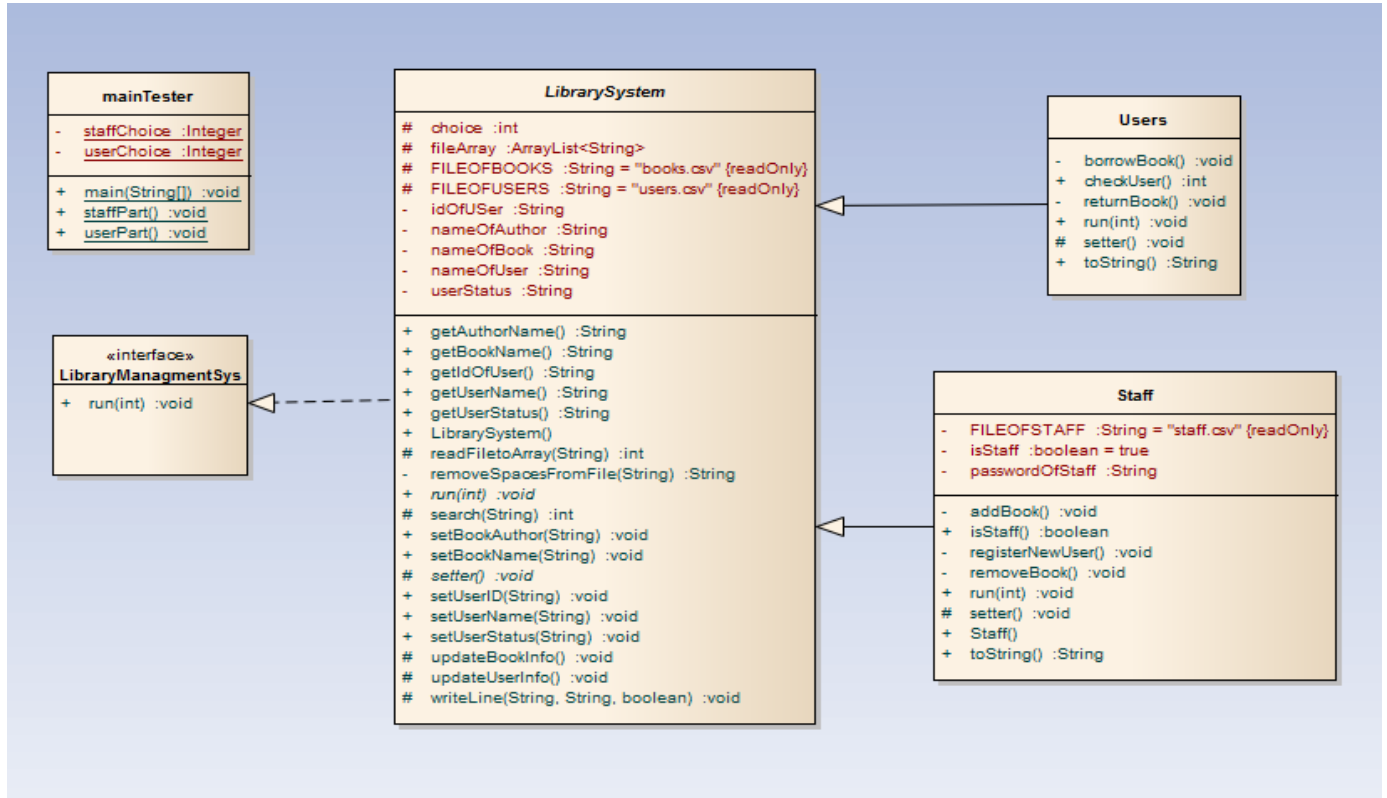
- Sistem kullanıcıları user ve staff olarak ikiye ayrılır.
- Sistemin user' lar düzgün bir şekilde çalışması için ilk önce userlar aşağıdaki özelliklerle eklenmeli
  - User name
  - User ID
- Ayrıca user'ların kütüphaneden kitap alıp geri getirme işlemlerini yapabilme işlemlerini gerçekleştirmek içinde kütüphaneye aşağıdaki özellikler ile kitap eklenmeli
  - Name of the book
  - Name of the author of the book
- System staff için ilk olarak çalıştığında staff genel bir şifre belirlemelidir. Bu şifre ile daha sonradan tekrardan giriş yapılacaktır.
- Staff user ile ilgili işlemler yaptığında veya user sisteme giriş yaptığında yukarıda belirtilen özellikler sistem tarafından istenilecektir.
- Staff kitaplar ile ilgili işlemler yaptığında veya user kitap alıp-getirme işlemlerini yaptığında yukarıda belirtilen özellikler sistem tarafından istenilecektir.
- Staff kendini user olarak sisteme ekleyebilir.

Program çalıştığı zaman yukarıda belirttiğim gerekçeleri açık bir şekilde istemektedir.

## 2. Use Case Diagrams



### 3. Class Diagrams



### 4. Other Diagrams

No Need Other Diagrams.

### 5. Problem Solutions Approach

Elimizde ki problemi çözmek için aşağıda belirttiğim işlemleri gerçekleştirdim.

- İlk olarak içinde tek bir fonksiyon içeren **LibraryManagementSys** adında bir interface oluşturdum.
- Bu interface'i impement eden **LibrarySystem** adında bir abstract class oluşturdum.
- Daha sonrasında bu abstract class'dan **Users** ve **Staff** adında iki class extends ettim.

Sistem **Staff** olarak çalıştırıldığında sadece bir şifre ile çalışabilecek şekilde ayarladım. Ve bu şifre program ilk çalıştığında **staff** tarafından belirlenecek. Kütüphane kullanıcıları **user name** ve **ID** 'ye kitaplar ise **book name** ve **author name** 'e sahiptir. Bu değerler ile ilgili metodların çoğunu **LibrarySystem** adlı class'da yazdım.

### 6. Test Cases

Oluşturulan **Library Management** sisteminde öncelikle **staff** olarak giriş yapılması gerekiyor. Ve **staff** olarak giriş yapıldıktan sonra kullanıcı(**users**) ve kitapların(**book**) **staff** tarafından eklenmesi gerekiyor. Bu işlemlerden sonra kayıtlı kullanıcılar kayıtlı kitapları alıp(**borrow**) geri getirebiliyorlar(**return**). Her kullanıcının 1 kitap alma hakkı var ikinci kitap için aldıkları kitabı geri getirmelidirler.

- Staff** olarak giriş yapıldı.
- Staff** olarak genel bir şifre belirlendi.
- Kullanıcılar(**users**) ve kitaplar(**books**) kütüphaneye eklendi.
- Kullanıcı(**user**) olarak giriş yapıldı.
- Kullanıcı kitap aldı(**borrow**).
- Kullanıcı kitabı geri getirdi(**return**).

-Belirlenen şifre ile staff kütüphaneden kitap kaldırabilir.

## 7. Running and Results

## 8.)Q2

Daha önce yapmış olduğum bazı denemeler sonucunda C de ADT yapısının aşağıdaki şekilde işlediği kararına vardım.

- Öncelikle iki dosyada çalıştım (.h ve .c)
- .h dosyasında bir structure yapısı oluşturdum. Ve bu structure yapısında bir pointerı parametre alan fonksiyonlar yazdım. Örneğin structure içinde int yapısında variablelar var ise bu variable ları set eden fonksiyonlar yazdım.
- .c dosyasında ise main fonksiyonda bu struct tipinde variablelar oluşturdum ve bu variable ların içindeki int degerlerini yazmış olduğum fonksiyonlar ile set ettim. Ve buna benzer bazı işlemler yaptım. Tabiki bu işlemleri yapmak için .h dosyasını include ettim.

Bu şekilde C de ADT yapısında kod yazmış oldum.

Structure yapısı ve pointer lar sayesinde bu ADT yapısı kolay bir hal aldı.