## horizontal line



LibAdmin

27.04.2017

**─**

Magada Mushabalo Craig

3239197

# Table of index

**Contents**

[horizontal line 0](#_gjdgxs)

[Table of index 1](#_3znysh7)

[Problem Background 4](#_3dy6vkm)

[Goals 4](#_1t3h5sf)

[Scope 4](#_4d34og8)

[Specifications 4](#_2s8eyo1)

[Project Members 5](#_3rdcrjn)

[**Name** 6](#_26in1rg)

[**Student Number** 6](#_lnxbz9)

[**Role** 6](#_35nkun2)

[Solution Background 6](#_44sinio)

[Technical Background 6](#_2jxsxqh)

[Solution Concept 6](#_z337ya)

[Risk Assessment 7](#_3j2qqm3)

[Project Plan 7](#_1y810tw)

[Task Allocation 7](#_4i7ojhp)

[Milestones 7](#_1ci93xb)

[Usecase Diagram 8](#_2bn6wsx)

[Class Diagram 9](#_3as4poj)

[Feasibility Assessment 10](#_1pxezwc)

[FEASIBILITY ANALYSIS 10](#_49x2ik5)

[Reference 10](#_2p2csry)

Agile Methodology

*I decided to use Agile methodology due to the ability to allow for better quality through regular testing and the flexibility to change whilst requiring minimal work. Above all other points, the ability for agile development requirements to emerge and evolve, and the ability to embrace change (with the appropriate trade-offs). The active involvement, cooperation and collaboration make agile development much more enjoyable. Instead of big specs, I focused on requirements. Instead of lengthy status reports, I collaborated around a task-board discussing progress.*

Abstract

*LibAdmin is an application developed to handle basic housekeeping functions of a library. It is a Ill organized solution for a library. LibAdmin is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like students login .It also has a facility of admin login, through which the admin can monitor the whole system. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date. The librarian after logging into his account i.e. admin account can generate various reports such as student report, issue report and book report. Overall this project of ours is being developed to help the students as Ill as students of library to maintain the library in the best way possible and also reduce the human efforts.*

# Problem Background

I was tasked with designing and implementing a computer application for a library management system.

## Goals

The goal of this project is to design and implement a Computer-based Library System Application to help Organise, Manage and control library transactions. Through the application, users should be able to register/login, borrow/return books and administrators should be able to generate reports, add/delete books and should be able to have the overview of the whole system.

## Scope

* The System should have two portals, one for Students and the other for the Administrator
* Students should be able to Login/Register
* Students should be able to View/Borrow/Return Books
* Administrators should be able to View Reports, Add/Delete/View Books

## Specifications

*USER LOGIN*

* This feature used by the user to login into system
* The user selects whether they want to login as student or admin
* . They are required to enter user id and password before they are allowed to enter the system.
* The user id and password will be verified
* And if invalid id is there user is allowed to not enter the system.

***Functional requirements***

* The system must only allow user with valid id and password to enter the system
* The user must be able to logout after they’ve completed using system

*REGISTER NEW USER*

* This feature can be performed by all users to create account.
* User must register them self by supplying their personal information.

***Functional requirements-***

* System must be able to verify information
* System needs to be able to store the users information in a database which is using as backend.
* System must be able to delete information if information is wrong

*REGISTER NEW BOOK*

* This feature allows admin to add new books to the library.
* Admin enters book details

***Functional requirements***

* System must be able to enter number of copies into table.
* System must be able to delete a book .

*SEARCH BOOK*

* This feature is found in book maintenance part . You can search book based on book name , category or by author name.

***Functional requirements***

* System must be able to search the database based on select search type
* System must be able to filter book based on keyword entered
* System must be able to show the filtered book in table view

*ISSUE BOOKS AND RETURN BOOKS*

* This feature allows to issue and return books and also view reports of book issued.

***Functional requirements***

* System must be able to enter issue information in database.
* System must be able to update number of books.
* System must be able to search if book is available or not before issuing books
* System should be able to enter issue and return date information

*Non-functional requirements*

***Safety Requirements***

* The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup

***Security Requirements***

* I are going to develop a secured database for the university .Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, delete, append etc. All other users other than library staff only have the rights to retrieve the information about database.

***Hardware Constraints***

* The system requires a database in order to store persistent data. The database should have backup capabilities

Time Constraints

I received the Project on the 15.03.2017, Initiated it on the 25.03.2017 and completed it on the 20.07.2017. It took me 5 weeks to finish the project.

## Project Members

|  |  |  |
| --- | --- | --- |
| **Name** | **Student Number** | **Role** |
| Magada M C | 3239197 | Project Manager |

# 

# Solution Background

I decided to implement the project in Python because of its design philosophy emphasizes code readability, understandability and reusability which are part of the Software Engineering goals and also part of the main attributes of a Ill engineered software. Python also allows for native database interfacing and far more concise code.

For Database Management I decided to use SQLite it is an in-process library that implements a serverless and a zero-configuration transactional SQL Database engine. SQLite3 also does not need external/separate server since it reads and writes directly to ordinary disk files. SQLite also brings the benefit of only needing a single database file which natively supports cross-platform use.

I also decided to use a Command-Based Interface in order to reduce 3rd party dependence. Commandline is always effective and efficient which are one of the Interface Design Principles.

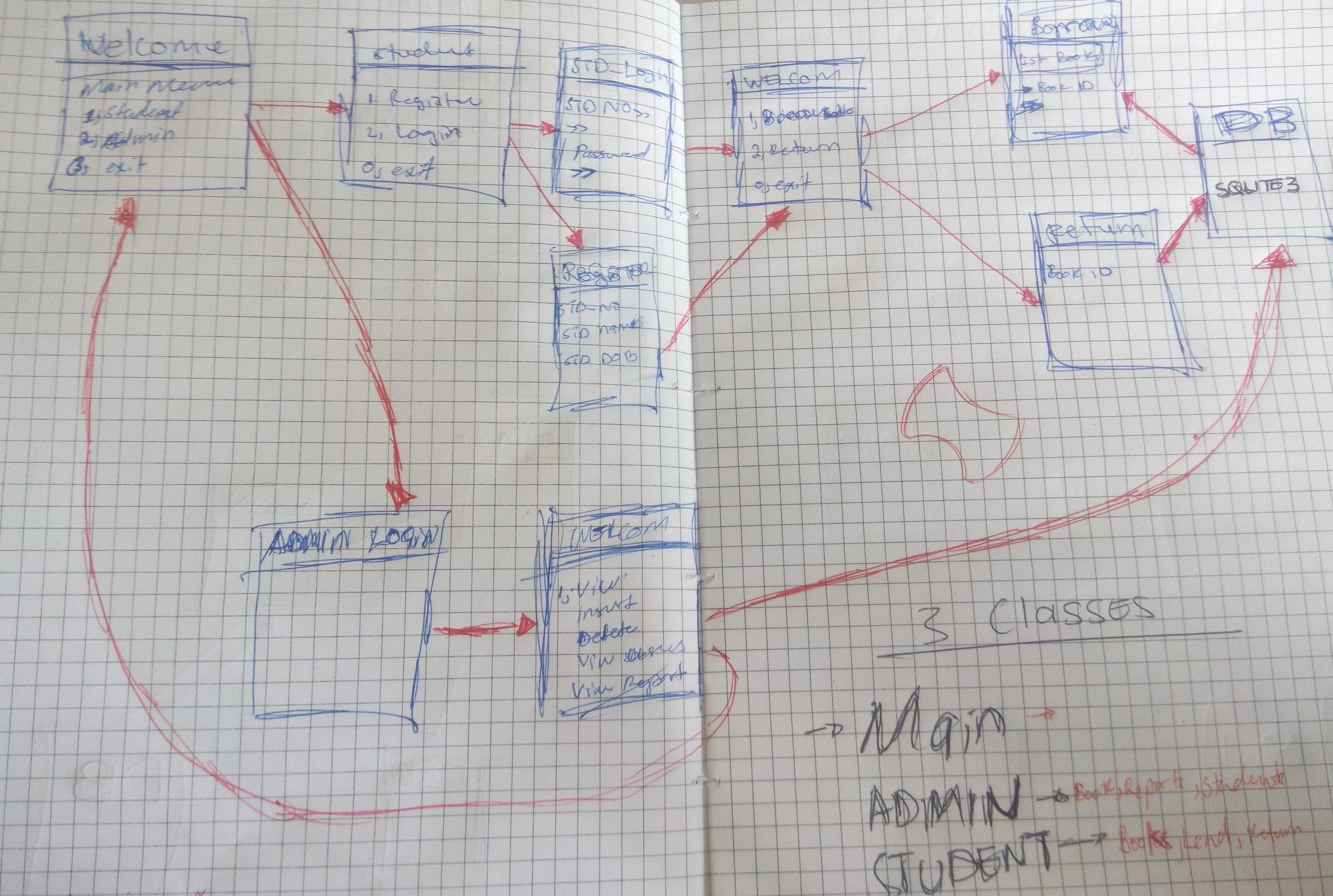
## Technical Background

Operating System - Mac OS, Linux, Microsoft Windows XP/Vista/7/10

3rd Party Software - Python 3, SQLite 3

Interface - Commandline Based

## Solution Concept



*Initial solution framework (red indicates program flow)*

# Risk Assessment

* Misinterpreting the problem and requirements
* Technological breakdown due to hardware/software failure
* Failure to define parameters and enforcing them

# Project Plan

## Task Allocation

|  |  |
| --- | --- |
| **Name** | **Task** |
| Magada Mushabalo Craig | PM/Python Coder |

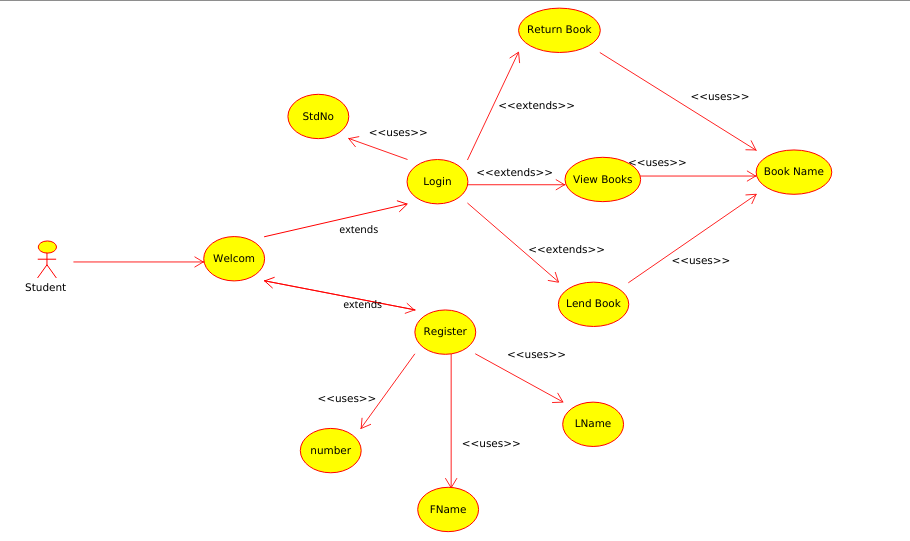
## 

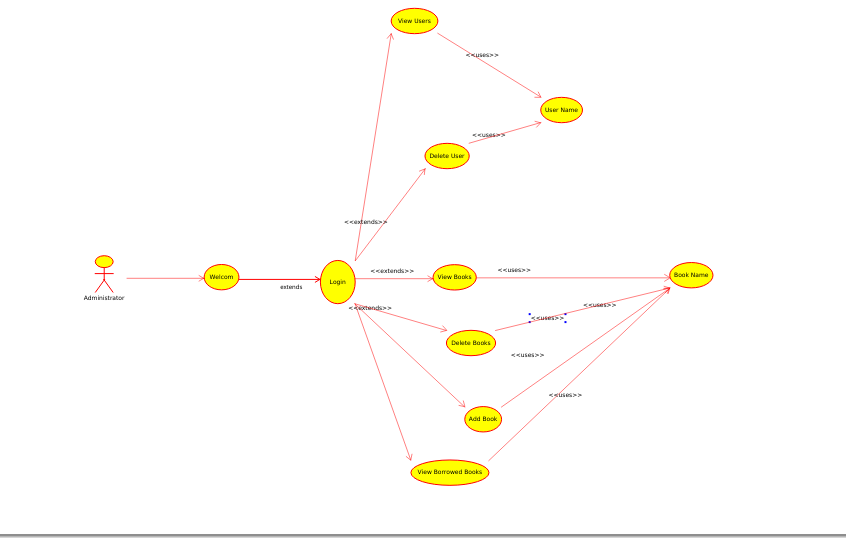
## Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Est date** | **Act date** | **Milestone** |
| V 1.0 | 25.03.2017 | 25.03.2017 | Installation of SQLite3  Coding Starts |
| V 1.1 | 01.04.2017 | 28.03.2017 | Adding the Register function  Creating a database tables |
| V 1.2 | 07.04.2017 | 04.04.2017 | Editing output text format  Added Exception Handling |
| v1.10 | 16.04.2017 | 10.04.2017 | Designing of Report output  Adding of wellcome/Login messages  Completed ahead of scheduled time |
| V 2 | 25.04.2017 | 20.04.2017 | Code Ready for presentation |

## 

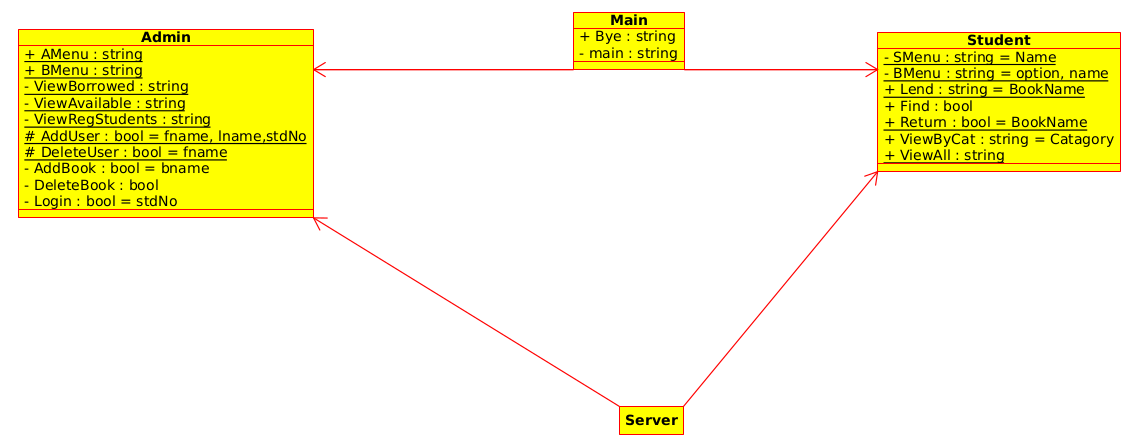
## Usecase Diagram





## 

## Class Diagram



# Feasibility Assessment

Our main trade off was that of the GUI, I decided that a command line based interface would present a bulletproof application that would not cause unnecessary complications in the morning when the project was due.

This idea was further reinforced when examining other groups’ projects where they Ire trying to balance both an external webserver and database in addition to their code. In addition, by the utilisation of the terminal environment, aside from the base Python distribution, the application can be run on any conventional platform. This prevents the need for server and 3rd party installations.

Our decision to employ SQlite was primarily due to its ease of setup, modularity and lack of scaling requirements. As the application would currently only be run at a single instance, scalability and user management wasn’t a concern.

## FEASIBILITY ANALYSIS

*Technical Feasibility:*

I can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as Ill. All the resources needed for the development of the software as Ill as the maintenance of the same is available. Here I are utilizing the resources which are available already.

*Economical Feasibility:*

Development of this application is highly economically feasible . The only thing is to be done is making an environment for the development with an effective supervision. If I are doing so, I can attain the maximum usability of the corresponding resources .Even after the development, I will not be in a condition to invest more in the organization .Therefore the system is economically feasible.

# Reference

1. Mcdonald, C. (2014). Https://www.dropbox.com/s/bwxhwldvyta6i3x/CSEIT\_2015\_Proceedings\_Paper\_18.pdf?dl=0. *6th Annual International Conference on Computer Science Education: Innovation & Technology (CSEIT 2015)*. doi:10.5176/2251-2195\_cseit15.30
2. Pressman, Roger S, and Bruce R Maxim. Software Engineering. New York: McGraw-Hill Higher Education, 2015. Print.
3. "About Sqlite". Sqlite.org. N.p., 2017. Ib. 26 Apr. 2017.
4. "How To Write A Science Project". *Science Buddies*. N.p., 2017. Ib. 26 Apr. 2017.
5. Pressman, Roger S, and Bruce R Maxim. Software Engineering. New York: McGraw-Hill Higher Education, 2015. Print.