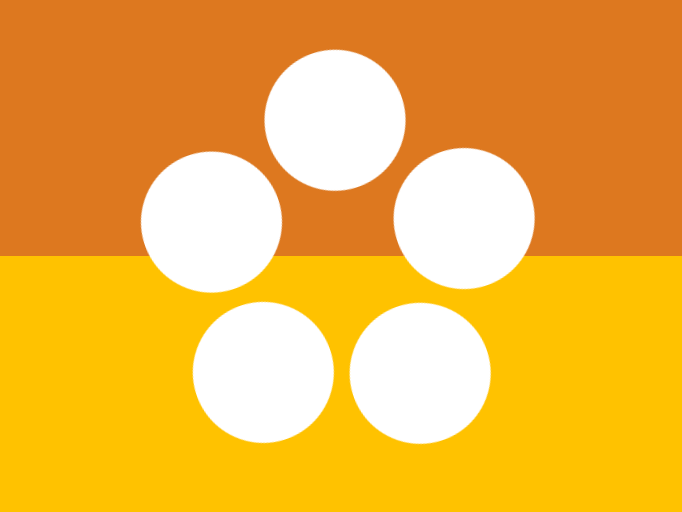
Proposal Analysis Report

*for the implementation of*

**Big Computer Corp Town Planner ©**

*in Tiny Town LGA*



Tiny Town LGA

Table of Contents

[Introduction 3](#_Toc4596567)

[**Information Systems Review of BCC Proposal** 4](#_Toc4596568)

[**Library System Database Prototype** 6](#_Toc4596569)

[Table Information 6](#_Toc4596570)

[Book Information Table – “Book” 6](#_Toc4596571)

[Members Information Table – “Member” 7](#_Toc4596572)

[Borrowings Information Table – “Borrow” 8](#_Toc4596573)

[Database Relationships 9](#_Toc4596574)

[Database Forms 9](#_Toc4596575)

Control of data assumed to be held by council, however a man in the middle attack could be performed (on the software side) for data mining purposes should BCC have malicious intent.

# Introduction

# **Information Systems Review of BCC Proposal**

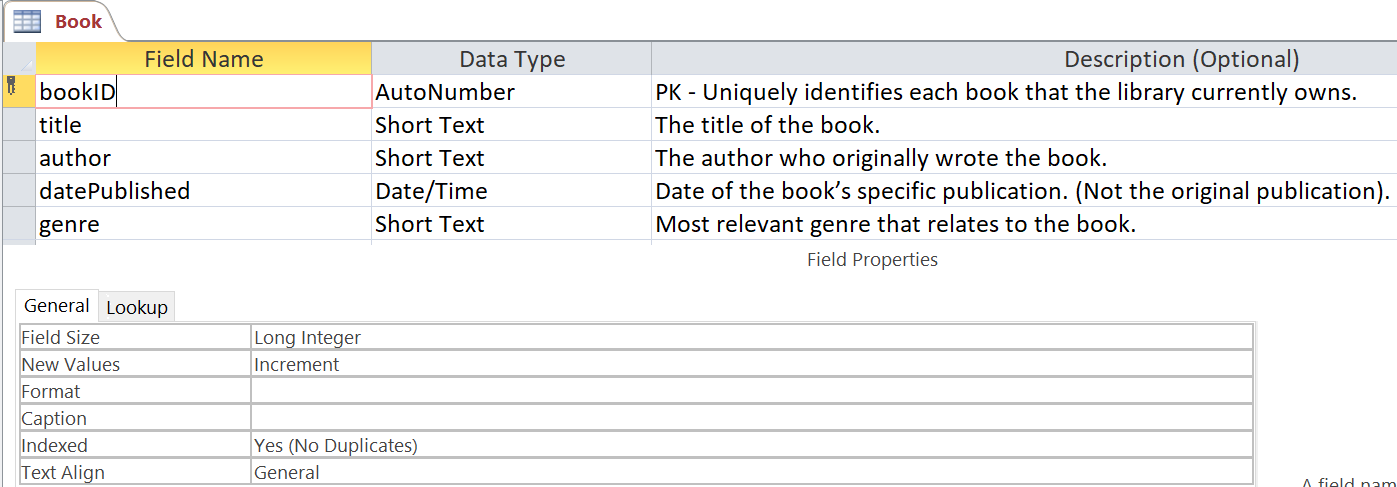
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Environment** | | Tiny Town LGA | | | |
| **Users** | | Tiny Town residents, Tiny Town library, Council employees | | | |
| **Purpose** | | 1. To provide access for residents of Tiny Town to view: rate notices, development applications, minutes from council meetings. 2. To allow residents to view (2.1 & 2.2) and update (2.3 & 2.4) personal details. 3. To allow council employees to send residents emails regarding overdue notices or upcoming events. 4. To allow residents to see library book lists, borrowed books, return dates, overdue books and any fines they may have accrued. | | | |
| **Information Processes** | | | | | |
|  | Process | | Participants | Data / Information | Information Tech |
| 1.1 | The Town Planner server retrieves information such as relevant rate notices, development applications (personal / local area) and video/audio from council meetings from an internal network council database using the resident ID number.  **(Storage/Retrieval, Transmitting/Receiving)** | | Resident (initiates the process) | Resident ID, resident details, rate notice slips, development application documents, video or audio from council meetings. | Internal council database system, Town Planner computer server, Network infrastructure |
| 1.2 | Transmit the information from 1.1 and then display the information to the resident’s Town Planner software package or web browser in a human readable format.  **(Analysing, Transmitting, Displaying)** | |  | Resident details, rate notice slips, development application documents, video or audio from council meetings. | PC, Browser, Town Planner computer server, Network infrastructure |
| 2.1 | The Town Planner server retrieves relevant personal details from the internal council database system.  **(Storage/Retrieval, Transmitting/Receiving)** | | Resident (initiates the process) | Resident ID, Resident details | Internal council database system, Town Planner computer server, Network infrastructure |
| 2.2 | Transmit the information from 2.1 and then display the information to the resident’s Town Planner software package or web browser in a human readable format.  **(Analysing, Transmitting/Receiving, Displaying)** | |  | Resident details | PC, Browser, Town Planner computer server, Network infrastructure |
| *2.3* | If the user updates the information displayed in 2.2, then send the new details to the Town Planner server.  **(Analysing, Transmitting, Displaying)** | | Resident (initiates the process) | Resident ID, Resident details | PC, Browser, Town Planner computer server, Network infrastructure |
| *2.4* | If the process in 2.3 occurred, update the records using the Resident ID with resident details data inside the council database system to be current with the residents updated details.  **(Storage/Retrieval, Transmitting/Receiving)** | |  | Resident ID, Resident Details | Internal council database system, Town Planner computer server, Network infrastructure |
| 3.1  OR | A council employee logs onto the software and creates an email to be sent. This is then submitted to the Town Planner computer server. (goes to 3.2)  **(Collecting, Transmitting/Receiving)** | | Council employee | Council department name, Message content, Recipient(s)’ details, | Town Planner computer server, Town Planner software, Network infrastructure, PC |
| 3.1 | Using the current date, the Town Planner computer server checks the internal council database for overdue notices or events and sends mail merged documents to 3.2.  **(Collecting, Analysing, Storage/Retrieval, Transmitting/Receiving, Processing)** | |  | Recipient(s)’ details, Current date | Town Planner computer server,  Internal council database system,  Network infrastructure |
| 3.2 | The Town Planner server receives the documents/messages from 3.1 and then sends emails to all recipient(s)’ email addresses.  **(Transmitting/Receiving, Displaying)** | |  | Council department name, Email message content, Recipient(s)’ details, | Town Planner computer server, |
| 4.1 | Retrieve a list of all available books from the library database, any borrowed books and their return date, overdue books and any fines that have been accrued by the resident.  **(Analysing, Storage/Retrieval, Transmitting/Receiving, Processing, Displaying)** | | Resident | Resident ID, Resident details, Book list, Overdue books and fines, return dates, Borrowed books list. | Library database, PC, Network infrastructure |

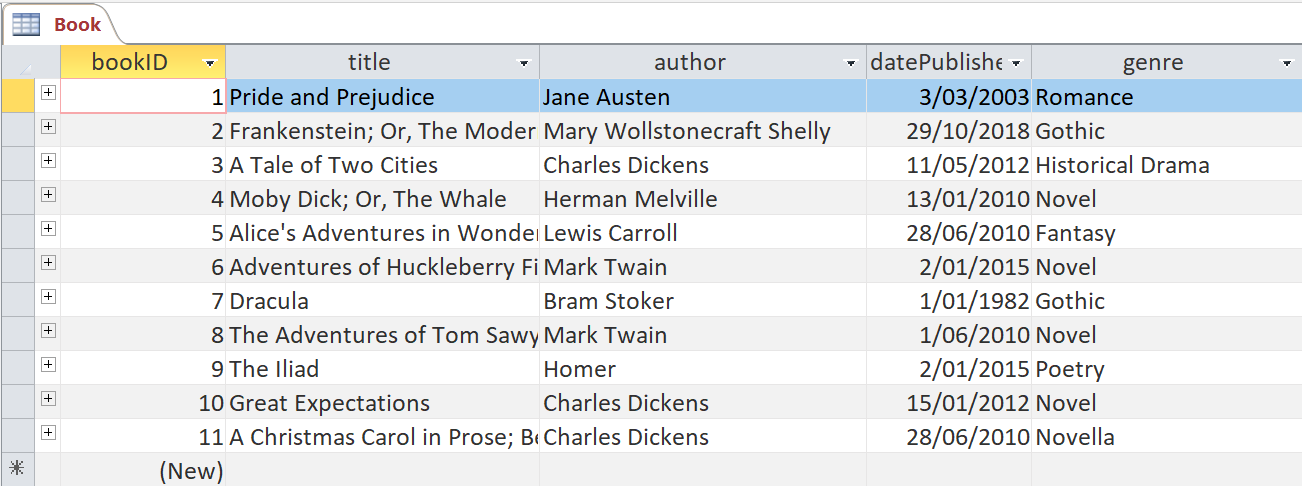
# **Library System Database Prototype**

## Table Information

### Book Information Table – “Book”

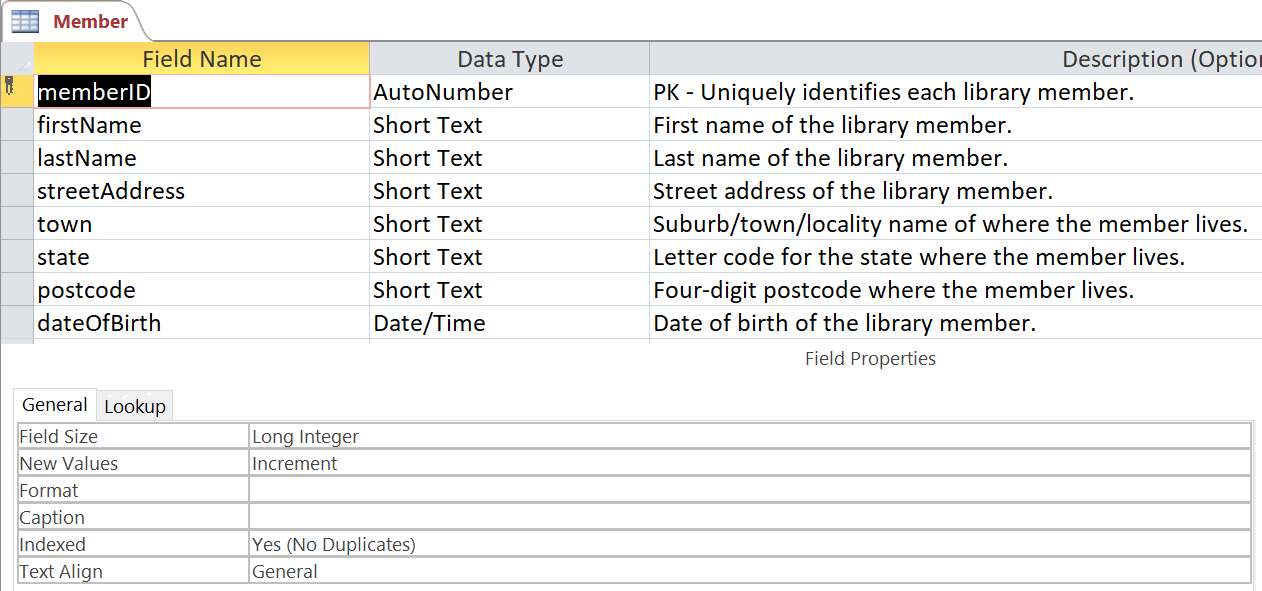
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field Name | Data Type | Length | Format | Description | Example |
| **PK** | bookID | Number | Int |  | Uniquely identifies each book that the library currently owns. |  |
|  | title | Text | 100 |  | A text field containing the title of the book. |  |
|  | author | Text | 100 |  | A text field containing the author who originally wrote the book. |  |
|  | datePublished | Date |  | dd/mm/yyyy | A date field containing the date of the book’s specific publication. (Not the original publication). |  |
|  | genre | Text | 50 |  | A text field containing the most relevant genre that relates to the book. |  |

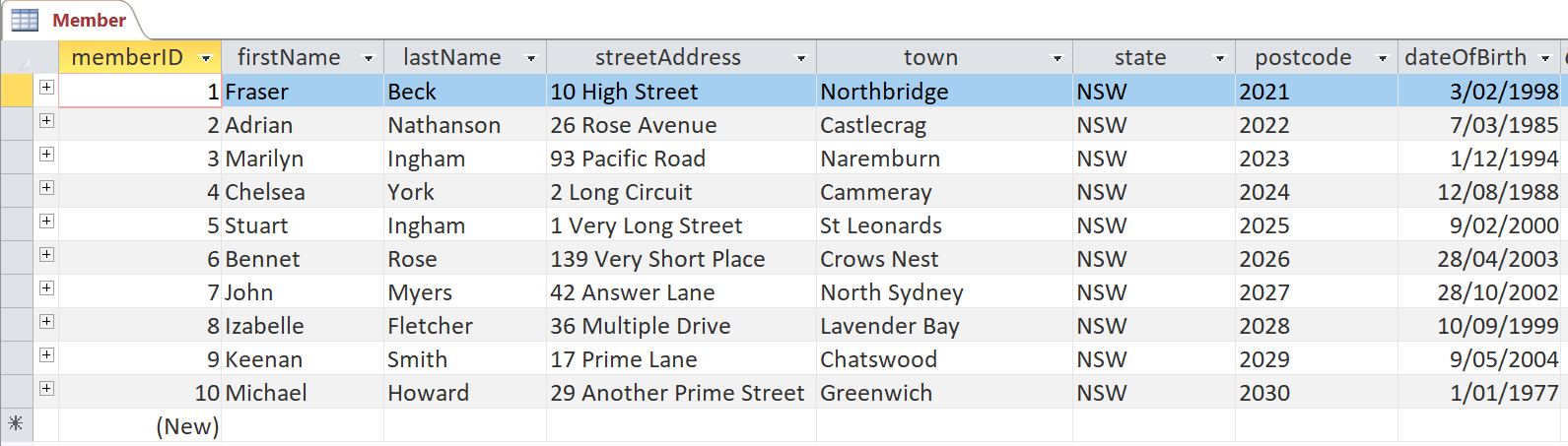




### Members Information Table – “Member”

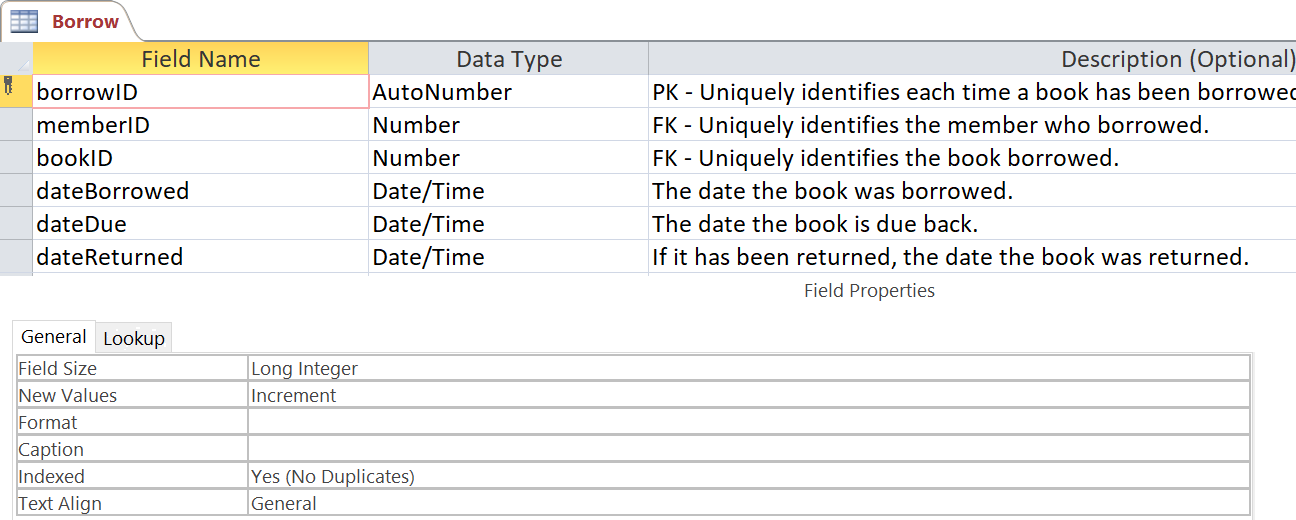
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field Name | Data Type | Length | Format | Description | Example |
| **PK** | memberID | Number | Int |  | Uniquely identifies each member of the library system. |  |
|  | firstName | Text | 100 |  | A text field containing the first name of the library member. |  |
|  | lastName | Text | 100 |  | A text field containing the last name of the library member. |  |
|  | streetAddress | Text | 100 |  | A text field containing the street address of the library member. |  |
|  | town | Text | 100 |  | A text field containing the suburb/town/locality name of where the member lives. |  |
|  | state | Text | 3 |  | A text field containing the letter code for the state where the member lives. |  |
|  | postcode | Text | 4 |  | A text field containing the four-digit postcode where the member lives. |  |
|  | dateOfBirth | Date |  | dd/mm/yyyy | A date field containing the date of birth of the library member. |  |

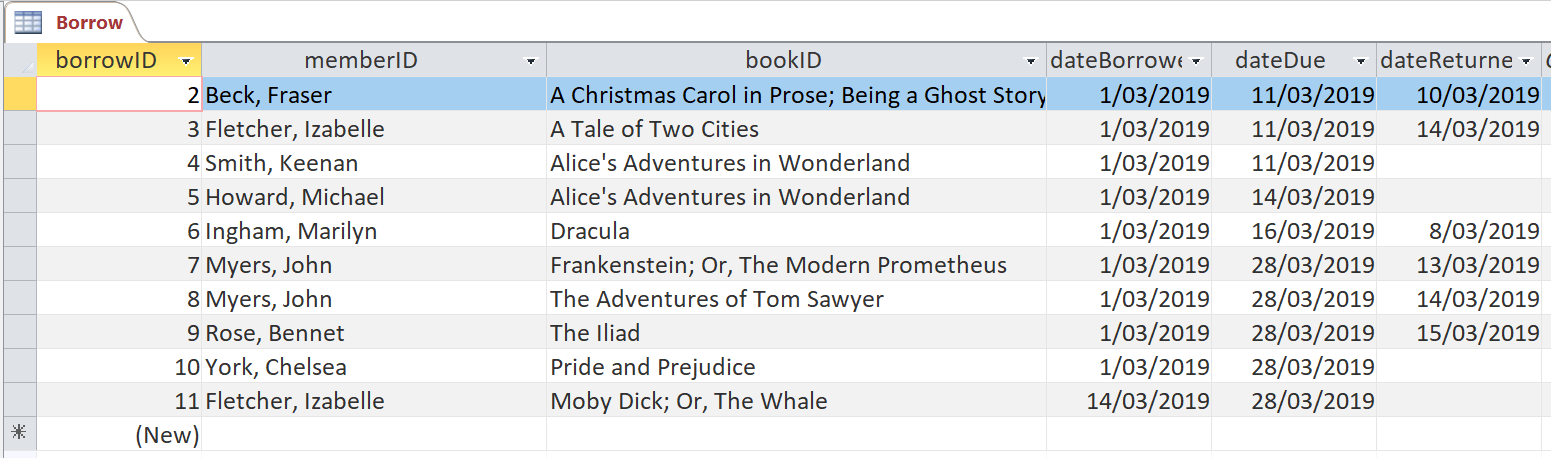




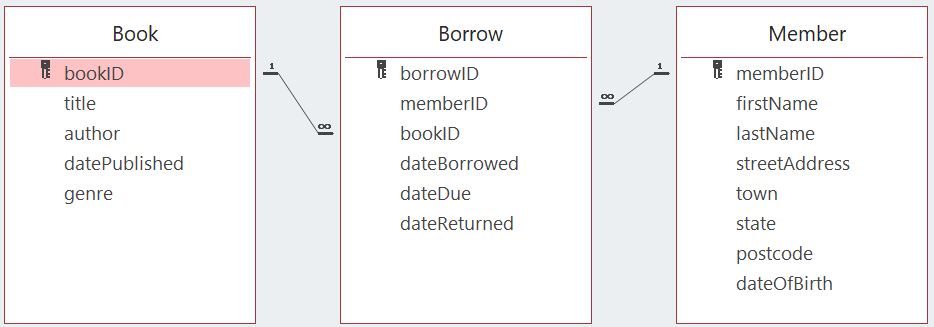
### Borrowings Information Table – “Borrow”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Field Name | Data Type | Length | Format | Description | Example |
| **PK** | borrowID | Number | Int |  | Uniquely identifies each time a book has been borrowed. |  |
| **FK** | memberID | Number | Int |  | Uniquely identifies the member which took out this borrowing. |  |
| **FK** | bookID | Number | Int |  | Uniquely identifies the book which was borrowed. |  |
|  | dateBorrowed | Date |  | dd/mm/yyyy | A date field containing the date when the book was borrowed. |  |
|  | dateDue | Date |  | dd/mm/yyyy | A date field containing the date when the book is due back. |  |
|  | dateReturned | Date |  | dd/mm/yyyy | A date field. If the book was returned then this will contain a date, otherwise this is null. |  |





### Database Relationships



## Database Forms