**TDD for Building Maintenance Corporation**

**DOCUMENT REVISION HISTORY**:

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| Version 1.0 | 13/03/2017 |  |

**TEAM SIZE**: 7

**TEAM MEMBERS**:

AARUSHI BHOLA

ANITA MOHANTY

ANURAG SAMANTA

DEV SOURAV DAS

DIVYA MALHOTRA

PRAMIT KUMAR PANIGRAHY

RUPALI GERA

**Table of Content**

**1. Introduction**

1.1. About the Project

1.2. Objective

1.3. Assumption

**2. General Description**

2.1. Tools used

2.2. General Constraints

**3. Design Details**

3.1. Architecture

3.1.1. High Level Design

3.1.2. Low Level Design

3.2. Report Specification

3.3. Conclusion

1. **Introduction**

* 1. **About the Project**

A program will be created which will use the payment records in the Flat Owners

Payments File to update the Amount–Paid field in the Flat Owner Master File and which

will then use the Flat Owner Master File to produce a report showing the fees outstanding.

**1.2 Objective**

Create a COBOL program which will apply a transaction file of Flat Owner Payments to the Flat Owner Master File which will then produce a report showing those flat owners whose fees are partially or wholly outstanding.

**1.3 Assumption**

The files Flat Owner Payments and the Flat Owner Master file are assumed to be initialized with 20 records.

**2. General Description**

**2.1. Tools used**

* VSAM Files
* COBOL
* JCL

**2.2. General Constraints**

* The program created was modular, with proper comments and adhered to the cobol coding standards.
* Copybooks were used in the program.
* Minimum of twenty records were created for the test data.
* All the reports had 20lines/page.
* All numeric data items in the report were edited.
* Data validation techniques were used.

**3. Design Details**

**3.1. Architecture**

**3.1.1. High Level Design (HLD)**

It explains the architecture that would be used for developing a software product. The

architecture diagram provides an overview of an entire system, identifying the main

components that would be developed for the product and their interfaces.

The HLD uses possibly nontechnical to mildly technical terms that should be

understandable to the administrators of the system. This document is can be used as a

reference manual for how the modules interact at a high level.

Flat-Owner-Payment-File Flat-Owner-Master-File

Master File

Transaction File

Outstanding Report File

**3.1.2. Low Level Design (LLD)**

It is a component-level design process that follows a step-by step refinement process.

This process can be used for designing data structures, required software architecture,

source code and ultimately, performance algorithms. Overall, the data organization may

be defined during requirement analysis and then refined during data design work. Post-

build, each component is specified in detail.

A copybook was created for both the Flat Owner Payment File and Flat Owner Master File.

**COBOL PROGRAM-1**

START

READ PAYMENTS FILE

READ OWNER MASTER FILE

IF READ IS SUCCESSFULL

INCREMENT FLAT OWNER NUMBER

NO

IF NAME MATCHES

YES

NO

UPDATE

AMOUNT

UPDATE OWNER NAME AND AMOUNT

UPDATED FILE-2

STOP

**ALGORITHM-1**

**STEP: 1**

**Both the Flat-Owner-Number and Flat-Owner-Payment-File are sorted in ascending order on the**

**basis of Flat-Owner-Number.**

**STEP: 2**

**Then Flat-Owner-Number** from **Flat-Owner-Payment-File** and **Flat-Owner-Number** from **Flat-**

**Owner-Master-File** are compared sequentially.

**STEP: 3**

If both the values above are found to be **equal** then

The **Flat-Owner-Name** of both the files is **checked**.

**STEP: 4**

**If** the **Flat-Owner-Name** in both the files is **equal** then

**Amount-Paid** in the **Flat-Owner-Master-File** is updated as:

**Step 4.1:** Initially the value of **Amount-paid** is **0**.

**Step 4.2:** Then, **Amount-paid** = **Amount-paid** + **Payment Amount.**

**Else**

**Step 4.3:**  **Rewrite** the **Owner- Name** in the **Flat-Owner-Payment File**.

**Step 4.4:** **Update** the **Amount-Paid** as per the above logic.

**STEP: 5**

Once the **Master File** is updated, it is further used for Report Generation.

**COBOL PROGRAM-2**

FILE CONTAINING AN ARRAY

READ UPDATED OWNER MASTER FILE

CALCULATE

FUNCTIONS

GENERATE

REPORT

**ALGORITHM-2**

**STEP: 1**

We have to create a FILE where we declare and initialize an **Array** consisting of :

Flat-Floor

Cost-per-Sqft

**STEP: 2**

In the same program, we read the **Updated Flat-Owner-Master-File** and using the **Search** verb, We compare Floor and Sq-ft of Flat-Owner-Master File with the Floor and Cost-Per-Sq-ft in the Array.

**STEP: 3**

Based on the Floor and its Cost-per-sq-ft, we calculate the **Total-Amount** that needs to be paid for maintenance for each Flat Owner. For Eg.-

If **Flat Floor** = 1 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else If **Flat Floor** = 2 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else if **Flat Floor** = 3 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else If **Flat Floor** = 4 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else If **Flat Floor** = 5 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else If **Flat Floor** = 6 then

Total **Amount** = Cost-per-Sqft \* Square-ft

Else

Display **‘Invalid Floor’**

End-if.

**STEP: 4**

Then Outstanding Amount for respective Flat-Owner is Calculated:

**Outstanding Amount** = **Total-Amount** – **Amount-paid**

**STEP: 5**

After Calculating the **Outstanding-Amount**, We can create a **PS file** with fields such as

Flat-Owner-Number

Flat-Owner-Name

Gender

Course-Code

Total-Amount

Amount-Paid

Outstanding-Amount

And generate the Report which is sorted in ascending order according to flat owner name.

.

**3.2. Report Specification**

The Fees field is a currency value with no digits after decimal point. The field should have

comma insertion and a floating dollar sign. The Amount Paid, Amount Outstanding , Total

Outstanding fields are currency values with floating dollar signs, comma insertion and two digits

after the decimal point.

**3.3.Conclusion**

The above algorithms were executed which used the payment records from the Flat-Owner- Payment-File to update the Amount field in the Flat-Owner-Master-File and which then used the Flat-Owner-Master-File to produce the Report.

The report will generate in the following format –

**BUILDING MAINTENANCE REPORT**

|  |  |
| --- | --- |
| **FLAT OWNER NUMBER :** |  |
| **FLAT OWNER NAME :** |  |
| **FLAT OWNER :** |  |
| **TOTAL AMOUNT :** |  |
| **AMOUNT PAID :** |  |
| **OUTSTANDING AMOUNT :** |  |
| **BALANCE :** |  |























