





## 1.2 Steps to Run the Program

*//contains the steps for running the application starting from the first run until all menus are tried to run*

The following are the steps for running the application

1. This application consists of various menus, mainly login, register, main, and account information. When the user login the first time the login menu will also ask for the user information, afterward the user doesn't need to input their information again.

```
=====
||                               LOGIN                               ||
=====
If you don't already have an account please type "register" in
the username input, if you want to quit type "quit" in the
username input
Username: register
=====
||                               REGISTER                               ||
=====
Enter your username: haqi
Enter your password: haqi
```

```
=====
||                               LOGIN                               ||
=====

If you don't already have an account please type "register" in
the username input, if you want to quit type "quit" in the
username input
Username: haqi
Password: haqi
Enter your name: Muhammad Baihaqi Aulia Asy'ari
Enter your phone number: 082336750134
Please enter your ID card number in this format
example: 3573052004691337
Enter your ID card number: 3573050101040001
Enter your salary: 8000000
Are you sure (y/n): y
Muhammad Baihaqi Aulia Asy'ari
082336750134
3573050101040001
8000000
```

```
=====
||                               MENU                               ||
=====

1. Credit card menu
2. Loan menu
3. Account information
4. Log out
5. Quit the program
menu: 3
```

```

=====
||                                ACCOUNT INFO                                ||
||=====||
||                                                                            ||
||          Name          : Muhammad Baihaqi Aulia Asy'ari                ||
||                                                                            ||
||          Phone number   : 082336750134                                ||
||                                                                            ||
||          ID card number : 3573050101040001                             ||
||                                                                            ||
||          salary         : 80000000                                     ||
||                                                                            ||
||=====||
Exit?
Are you sure (y/n): |

```

2. Some menus can be explored, unfortunately not all intended features are included. This is due to the time constraint and limitation on what's allowed to be used in Java for this project. The user will find out that any child directories of the "Credit card menu" will always result in "not found" and immediately redirected to the parent directory.

```

=====
||                                CREDIT CARD                                ||
||=====||
1. Apply for a credit card
2. Owned Credit card
3. Back to main menu
menu: 1

```

```

=====
||                                APPLY FOR A CREDIT CARD                    ||
||=====||
1. General purpose
2. Travel
3. Lifestyle
4. Priority
5. Back to credit card menu
menu: 1

```

```

=====
||          GENERAL PURPOSE CREDIT CARD APPLICATION          ||
=====

  _ _ _ _ _
| | | | / _ \ | | | | | |
| | | | | | | | | |
| _ _ _ | | | | _ _ _ |
    | | | | | | |
    | _ | \ _ _ /   | _ |

  _ _ _ _ _
| \ | | _ _ _ | | _ / _ | _ _ _ _ _ | | | |
| : | / _ \ | | | _ / _ \ | | | | ' \ / _ ` |
| _ \ _ \ _ _ / \ _ | | _ \ _ _ / \ _ | | | | \ _ _ , _ |

=====
||          APPLY FOR A CREDIT CARD          ||
=====

1. General purpose
2. Travel
3. Lifestyle
4. Priority
5. Back to credit card menu
menu:

```

3. The flaw above is also exhibited by “Apply for a loan” in the “Loan menu” for all loans except for “Mortgage”.

```

=====
||          LOAN          ||
=====

1. Apply for a loan
2. Current loan status
3. Back to main menu
menu: 1

```

```
=====
||                               PERSONAL LOAN                               ||
=====

  _ _ _ _ _
| | | | / _ \ | | |
| | | | _ | | | | | _
| _ _ _ | | | | _ _ _ | |
    | | | | _ | | |
    | _ | \ _ _ /   | _ |

  _ _ _ _ _
| \ | | _ _ _ | | _ / _ | _ _ _ _ _ _ _ _ _ | | | | | |
| . ^ | / _ \ | _ | | _ | / _ \ | | | | ^ \ / ^ |
| _ | \ | \ _ _ / \ _ | | _ | \ _ _ / \ _ , | | | | | \ _ , |

=====
||                               APPLY FOR A LOAN                               ||
=====

1. Personal
2. Auto
3. Mortgage
4. Refinancing
5. Back to loan menu
menu:
```

4. Although, some features are not what they'd seem to be. One function is purposely made to print "not found" if another function has not been used. This is the "Current loan status" menu, where it only outputs something besides "not found" if the user has made a "mortgage simulation" from the "Mortgage" menu in "Apply for a loan".

```

=====
||                                LOAN                                ||
=====
1. Apply for a loan
2. Current loan status
3. Back to main menu
menu: 3
=====
||                                MENU                                ||
=====
1. Credit card menu
2. Loan menu
3. Account information
4. Log out
5. Quit the program
menu: 2
=====
||                                LOAN                                ||
=====
1. Apply for a loan
2. Current loan status
3. Back to main menu
menu: 2
=====
||                                LOAN STATUS                        ||
=====

  _ _ _ _ _
| | | / _ \ | | |
| | | _ | | | | | _
| _ _ | | | | _ _ |
  | | | _ | | |
  | _ \ _ _ / | _ |

  _ _ _ _ _
| \ | _ _ | _ / _ | _ _ _ _ _ | | |
| : | / _ \ | _ | _ / _ \ | | ' \ / _ |
| _ \ \ _ _ / \ _ | _ | \ _ _ / \ , _ | | | \ _ , _ |

Exit?Are you sure (y/n):

```



5. Like what has previously been said, the “Mortgage” menu is the only one in the “Apply for a loan” that is functional. To use the “Mortgage” menu or the “mortgage simulation” the user must input the purpose of the Mortgage.

```
=====
||                               MORTGAGE LOAN                               ||
=====
Purpose of Credit
1. Buying a house
2. Renovating
menu:
```

6. When entering the buying menu, the user will be prompted with collateral type that “Renovation“ doesn’t have. After entering the collateral type the user is again prompted with the building Area. Every decision here is adding a minimum down payment percentage in the background. The last consideration for the minimum down payment percentage is the credit facility which counts the overall credit you own. But for this case, we can enter anything. But entering 2 or 3+ will increase the minimum down payment percentage by up to 10%.

```
=====
||                               MORTGAGE LOAN                               ||
=====
Purpose of Credit
1. Buying a house
2. Renovating
menu: 1
Collateral Type
1. House
2. Apartment
3. Shop
menu: 1
Building Area (m2): 70
How many Credit Facilities do you have
Credit Facility: 1
```

7. After entering the value to be considered for the down payment percentage the user is asked for the length of tenor for the mortgage which will affect the calculated max proposed mortgage accounting for the minimum down payment percentage previously

considered in the background. Afterward, the user is asked for their proposed mortgage and down payment. If the down payment proposed by the user is higher than the minimum down payment amount, the user is allowed to readjust their proposed mortgage if needed.

```
Building Area (m2): 70
How many Credit Facilities do you have
Credit Facility: 1
Maximum 20 years tenor
Tenor: 15
Maximum proposed debt: 535,555,667
House price: 450000000
Minimum down payment amount: 45,000,000
Down payment: 500000000
Maximum proposed debt: 532,000,100
House price: 450000000
```

8. Next, a table will display the information the user has entered and will ask if the user is sure about the proposed mortgage, if they are, the data will be stored and the user will be directed to the parent menu, if they aren't so sure, the user will have the chance to re-enter the information again.

```

=====
||                               Mortgage Application                               ||
||=====||
||
||      Credit facility      : 1st
||
||      House price         : IDR 450,000,000
||
||      Down payment 11.11% : IDR 50,000,000
||
||      Debt principal      : IDR 400,000,000
||
||      Tenor               : 15
||
||      Interest            : 7.25%
||
||      installment         : IDR 3,651,451.52
||
||      Minimum income      : IDR 6,639,002.77
||
||=====||
Are you sure (y/n):

```

9. After the user is assured of their own decision, the user could see the information about their loan in the “Current loan status” menu.

```

=====
||                          APPLY FOR A LOAN                          ||
=====
1. Personal
2. Auto
3. Mortgage
4. Refinancing
5. Back to loan menu
menu: 5
=====
||                          LOAN                                      ||
=====
1. Apply for a loan
2. Current loan status
3. Back to main menu
menu: 2
=====
||                          LOAN STATUS                              ||
=====
=====
||                          Mortgage Application                      ||
||=====||
||                                                                    ||
||                                                                    ||
||          Credit facility      : 1st                               ||
||                                                                    ||
||          House price          : IDR 450,000,000                   ||
||                                                                    ||
||          Down payment 11.11% : IDR 50,000,000                     ||
||                                                                    ||
||          Debt principal       : IDR 400,000,000                   ||
||                                                                    ||
||          Tenor                : 15                                ||
||                                                                    ||
||          Interest             : 7.25%                             ||
||                                                                    ||
||          installment          : IDR 3,651,451.52                 ||
||                                                                    ||
||          Minimum income       : IDR 6,639,002.77                 ||
||                                                                    ||
||=====||
Exit?Are you sure (y/n):

```

10. That is as far as the program goes, to quit the program the user could enter the quit menu in the main menu or log out and type quit.

```
=====
||                               LOAN                               ||
=====
1. Apply for a loan
2. Current loan status
3. Back to main menu
menu: 3
=====
||                               MENU                               ||
=====
1. Credit card menu
2. Loan menu
3. Account information
4. Log out
5. Quit the program
menu: 5
=====
||                               QUIT SUCCESSFULLY                     ||
=====

Process finished with exit code 0
```

```
=====
||                               LOGIN                               ||
=====
If you don't already have an account please type "register" in
the username input, if you want to quit type "quit" in the
username input
Username: quit
=====
||                               QUIT SUCCESSFULLY                     ||
=====

Process finished with exit code 0
```

### **1.3 Program Code**

//contains the code of all the programs that have been created

```

1  package com.baihaqi.bankingcreditcli;
2
3  import java.util.Scanner;
4
5  /**
6   *
7   * @author G4CE-PC
8   *      Muhammad Baihaqi Aulia Asy'ari
9   *      2241720145 - TI 1I - 19
10  */
11  public class BankingCreditCLI {
12      final static Scanner input = new Scanner(System.in);
13      static String[] [] credential = new String[1][2];
14      static String username;
15      static double[] [] creditMortgage = new double[1][4];
16      static String[] creditMortgageDetail = new String[1];
17      static String[] [] profile = new String[1][4];
18
19      public static void main(String[] args) {
20          credential[0][0] = "admin";
21          credential[0][1] = "admin";
22          loginMenu();
23      }
24
25      // region login
26      static String usernameCheck() {
27          while (true) {
28              write("Username: ");
29              String userInput = input.next();
30              for (String[] strings : credential) {
31                  if (strings[0] == null)
32                      continue;
33                  if (strings[0].equals(userInput))
34                      return userInput;
35                  if (userInput.equalsIgnoreCase("register"))
36                      return userInput;
37                  if (userInput.equalsIgnoreCase("quit"))
38                      return userInput;
39              }
40              printPromptSplit(
41                  "The username you've entered doesn't exist in our system please
↪ re-enter your username correctly");
42          }
43      }
44
45      static boolean passwordCheck() {
46          int limit = 0;
47          while (limit < 3) {
48              write("Password: ");
49              String userInput = input.next();
50              for (String[] strings : credential) {
51                  if (strings[0].equals(username)) {
52                      if (strings[1].equals(userInput)) {
53                          return true;

```

```

54         }
55     }
56 }
57 if (limit < 1)
58     writeln("Wrong password");
59 if (limit == 1)
60     writeln("""
61         Wrong password, Last attempt
62         if you fail again, you would need to re-enter your username""");
63     limit++;
64 }
65 return false;
66 }
67
68 static boolean attemptLogin() {
69     username = usernameCheck();
70     if (!(username.equalsIgnoreCase("register") ||
↪ username.equalsIgnoreCase("quit"))) {
71         boolean password = passwordCheck();
72         if (password) {
73             if (profile[getUserID()][0] == null)
74                 setProfile();
75             mainMenu();
76         } else {
77             if (attemptLogin()) {
78                 if (username.equalsIgnoreCase("register"))
79                     registerMenu();
80                 if (username.equalsIgnoreCase("quit"))
81                     quitMenu();
82             }
83         }
84         return false;
85     }
86     return true;
87 }
88
89 // endregion
90 // region print
91 static void printHeading(String prompt) {
92     int heading;
93     String bar;
94     String side = "||";
95     if (prompt.length() % 2 == 0) {
96         heading = 64;
97         bar = "=====";
98     } else {
99         heading = 65;
100         bar = "=====";
101     }
102     int gap = ((heading - (side.length() * 2) - prompt.length()) / 2);
103     String title = String.format("%s" + gap + "s%s" + gap + "s\n", side, " ",
↪ prompt, " ", side);
104     writeln(bar);
105     write(title);

```



```

106         writeln(bar);
107     }
108
109     static void printPromptSplit(String prompt) {
110         String[] promptSplit = prompt.split("\\s");
111         int i = 0;
112         while (i < promptSplit.length) {
113             int limit = 0;
114             while (limit < 65 && i < promptSplit.length && (limit +
↪ promptSplit[i].length()) < 65) {
115                 write(String.format("%s ", promptSplit[i]));
116                 limit = limit + (promptSplit[i].length() + 1);
117                 i++;
118             }
119             writeln("");
120         }
121     }
122
123     static void write(String s) {
124         System.out.print(s);
125     }
126
127     static void writeln(String s) {
128         System.out.println(s);
129     }
130
131     // endregion
132     // region etc
133     static void newStringArray(String[][] data) {
134         String[][] old = data;
135         data = new String[old.length + 1][old[0].length];
136         for (int row = 0; row < old.length; row++) {
137             for (int col = 0; col < old[row].length; col++) {
138                 data[row][col] = old[row][col];
139             }
140         }
141     }
142
143     static void newString(String[] data) {
144         String[] old = data;
145         data = new String[data.length + 1];
146         for (int i = 0; i < old.length; i++) {
147             data[i] = old[i];
148         }
149     }
150
151     static void newDoubleArray(double[][] data) {
152         double[][] old = data;
153         data = new double[data.length + 1][data[0].length];
154         for (int row = 0; row < old.length; row++) {
155             for (int col = 0; col < old[row].length; col++) {
156                 data[row][col] = old[row][col];
157             }
158         }

```

```

159     }
160
161     static String promptedTextInput(String prompt) {
162         write(prompt);
163         return input.next();
164     }
165
166     static double powerDouble(double base, int exponent) {
167         if (exponent == 0) {
168             return 1;
169         } else {
170             return base * powerDouble(base, exponent - 1);
171         }
172     }
173
174     static void notFound() {
175         writeln("""
176
177             _ _ _ _ _ \s
178             | | | / _ \ | | | \s
179             | | | _ | | | | | \s
180             | _ _ | | | | _ _ |
181             | | | _ | | | | \s
182             | _ | \ _ _ _ / | | \s
183
184                 \s
185                 \s
186
187             _ _ _ _ _ _ _ _ _ _ \s
188             | \ | | _ _ _ | _ _ / | _ _ _ _ _ _ _ _ _ _ |
189             | . / _ \ | _ | | / _ \ | | | | \ / _ ' |
190             | _ \ | \ _ _ _ / \ _ | | | \ _ _ _ / \ _ , _ | | | | \ _ _ , _ |
191
192                 \s""");
193     }
194
195     static boolean confirm() {
196         while (true) {
197             write("Are you sure (y/n): ");
198             String userInput = input.next();
199             if (userInput.equalsIgnoreCase("y")) {
200                 return true;
201             } else if (userInput.equalsIgnoreCase("n")) {
202                 return false;
203             }
204             writeln("Please enter a valid input!");
205         }
206     }
207
208     static int getUserID() {
209         int i = 0;
210         if (!credential[i][0].equals(username)) {
211             do {
212                 i++;
213             } while (!credential[i][0].equals(username));
214         }
215         return i;
216     }
217 }

```

```

213
214 static void insertMortgageDetail(String s) {
215     write(s + "\n");
216     creditMortgageDetail[getUserID()] += s + "\n";
217 }
218
219 static boolean validatePhoneNumber(String numbers) {
220     // 0895388899808 -> 13 digits
221     // 082336750134 -> 12 digits
222     // 08912888374 -> 11 digits
223     int length = numbers.length();
224
225     if (length < 11 || length > 13)
226         return false;
227
228     for (int i = 0; i < length; i++) {
229         // validate if each element is a number
230
231         int current = Integer.parseInt(String.format("%c", numbers.charAt(i)));
232         // first digit should be 0
233         if (i == 0 && current != 0)
234             return false;
235
236         // second digit should be 8
237         if (i == 1 && current != 8)
238             return false;
239     }
240     return true;
241 }
242
243 static boolean validateIDCardNumber(String numbers) {
244     // 3573051004040001
245     // 3573056204040001
246     int length = numbers.length();
247     if (length != 16)
248         return false;
249     for (int i = 0; i < length; i++) {
250         switch (numbers.charAt(i)) {
251             case '1', '2', '3', '4', '5', '6', '7', '8', '9', '0' -> {
252                 }
253             default -> {
254                 return false;
255             }
256         }
257     }
258     return true;
259 }
260
261 static int[] debtReadjustment(double installmentMin, double
↪ inverseReturnOfPoweredInterest, double interestInMonth, int downPayment, int
↪ downPaymentPercentage) {
262     boolean repeat = true;
263     double creditLimit;
264     double debtMax;

```

```

265     int debt;
266     int[] out = new int[2];
267     do {
268         creditLimit = installmentMin
269             * (inverseReturnOfPoweredInterest / interestInMonth);
270         debtMax = creditLimit + downPayment;
271         write(String.format("Maximum proposed debt: %.0f\n", debtMax));
272         write("House price: ");
273         debt = input.nextInt();
274         if (debt > debtMax) {
275             writeln("Please enter a value smaller than the maximum");
276         } else {
277             repeat = false;
278         }
279     } while (repeat);
280     if (downPayment < (double) (debt / 100) * downPaymentPercentage) {
281         write(String.format("Minimum down payment amount: %.0f\n", (double) (debt /
↪ 100) * downPaymentPercentage));
282         repeat = true;
283         do {
284             write("Down payment: ");
285             downPayment = input.nextInt();
286             if (downPayment < (debt / 100) * downPaymentPercentage) {
287                 writeln("Please enter a value bigger than the minimum!");
288             } else {
289                 repeat = false;
290             }
291         } while (repeat);
292     } else {
293         out[0] = debt;
294         out[1] = downPayment;
295         return out;
296     }
297     if (downPayment > (double) (debt / 100) * downPaymentPercentage) {
298         out = debtReadjustment(installmentMin, inverseReturnOfPoweredInterest,
↪ interestInMonth, downPayment, downPaymentPercentage);
299     }
300     out[0] = debt;
301     out[1] = downPayment;
302     return out;
303 }
304
305 // endregion
306 // region menu
307 static void loginMenu() {
308     printHeading("LOGIN");
309     printPromptSplit(
310         "If you don't already have an account please type \"register\" in the
↪ username input, if you want to quit type \"quit\" in the username input");
311     if (attemptLogin()) {
312         if (username.equalsIgnoreCase("register"))
313             registerMenu();
314         if (username.equalsIgnoreCase("quit"))
315             quitMenu();

```

```

316     }
317 }
318
319 static void registerMenu() {
320     printHeading("REGISTER");
321     newStringArray(credential);
322     newStringArray(profile);
323     newDoubleArray(creditMortgage);
324     newString(creditMortgageDetail);
325     creditMortgageDetail[creditMortgageDetail.length - 1] = "";
326     credential[credential.length - 1][0] = promptedTextInput("Enter your username:
↵ ");
327     credential[credential.length - 1][1] = promptedTextInput("Enter your password:
↵ ");
328     loginMenu();
329 }
330
331 static void setProfile() {
332     int id = getUserID();
333     String name, phoneNumber, IDCardNumber, salary;
334     input.nextLine();
335     write("Enter your name: ");
336     name = input.nextLine();
337     boolean i = true;
338     do {
339         write("Enter your phone number: ");
340         phoneNumber = input.next();
341         if (validatePhoneNumber(phoneNumber)) {
342             i = false;
343         } else {
344             writeln("Please enter a valid phone number");
345         }
346     } while (i);
347     writeln("Please enter your ID card number in this format");
348     writeln("example: 3573052004691337");
349     i = true;
350     do {
351         write("Enter your ID card number: ");
352         IDCardNumber = input.next();
353         if (validateIDCardNumber(IDCardNumber)) {
354             i = false;
355         } else {
356             writeln("Please enter a valid ID card number");
357         }
358     } while (i);
359     salary = promptedTextInput("Enter your salary: ");
360
361     if (confirm()) {
362         profile[id][0] = name;
363         profile[id][1] = phoneNumber;
364         profile[id][2] = IDCardNumber;
365         profile[id][3] = salary;
366         writeln(profile[id][0]);
367         writeln(profile[id][1]);

```

```

368         writeln(profile[id][2]);
369         writeln(profile[id][3]);
370     } else {
371         setProfile();
372     }
373 }
374
375 static void mainMenu() {
376     printHeading("MENU");
377     writeln("""
378         1. Credit card menu
379         2. Loan menu
380         3. Account information
381         4. Log out
382         5. Quit the program""");
383     switch (promptedTextInput("menu: ")) {
384         case "1" -> creditCardMenu();
385         case "2" -> loanMenu();
386         case "3" -> accountInfoMenu();
387         case "4" -> loginMenu();
388         case "5" -> quitMenu();
389     }
390 }
391
392 // region mainMenu
393 static void creditCardMenu() {
394     printHeading("CREDIT CARD");
395     writeln("""
396         1. Apply for a credit card
397         2. Owned Credit card
398         3. Back to main menu""");
399     switch (promptedTextInput("menu: ")) {
400         case "1" -> newCreditCard();
401         case "2" -> ownedCreditCard();
402         case "3" -> mainMenu();
403     }
404 }
405
406 // region creditCardMenu
407 static void newCreditCard() {
408     printHeading("APPLY FOR A CREDIT CARD");
409     writeln("""
410         1. General purpose
411         2. Travel
412         3. Lifestyle
413         4. Priority
414         5. Back to credit card menu""");
415     switch (promptedTextInput("menu: ")) {
416         case "1" -> generalPurposeCreditCardApplication();
417         case "2" -> travelCreditCardApplication();
418         case "3" -> lifestyleCreditCardApplication();
419         case "4" -> priorityCreditCardApplication();
420         case "5" -> creditCardMenu();
421     }

```

```

422     }
423
424     // region newCreditCard
425     static void generalPurposeCreditCardApplication() {
426         printHeading("GENERAL PURPOSE CREDIT CARD APPLICATION");
427         notFound();
428         newCreditCard();
429     }
430
431     static void travelCreditCardApplication() {
432         printHeading("TRAVEL CREDIT CARD APPLICATION");
433         notFound();
434         newCreditCard();
435     }
436
437     static void lifestyleCreditCardApplication() {
438         printHeading("LIFESTYLE CREDIT CARD APPLICATION");
439         notFound();
440         newCreditCard();
441     }
442
443     static void priorityCreditCardApplication() {
444         printHeading("PRIORITY CREDIT CARD APPLICATION");
445         notFound();
446         newCreditCard();
447     }
448
449     // endregion
450     static void ownedCreditCard() {
451         printHeading("OWNED CREDIT CARD");
452         notFound();
453         creditCardMenu();
454     }
455
456     // endregion
457     static void loanMenu() {
458         printHeading("LOAN");
459         writeln("""
460             1. Apply for a loan
461             2. Current loan status
462             3. Back to main menu""");
463         switch (promptedTextInput("menu: ")) {
464             case "1" -> newLoanMenu();
465             case "2" -> accountLoanInfo();
466             case "3" -> mainMenu();
467         }
468     }
469
470     // region loanMenu
471     static void newLoanMenu() {
472         printHeading("APPLY FOR A LOAN");
473         writeln("""
474             1. Personal
475             2. Auto

```

```

476         3. Mortgage
477         4. Refinancing
478         5. Back to loan menu""");
479     switch (promptedTextInput("menu: ")) {
480         case "1" -> personalLoanApplication();
481         case "2" -> autoLoanApplication();
482         case "3" -> mortgageLoanApplication();
483         case "4" -> refinancingLoanApplication();
484         case "5" -> loanMenu();
485     }
486 }
487
488 // region newLoanMenu
489 static void personalLoanApplication() {
490     printHeading("PERSONAL LOAN");
491     notFound();
492     newLoanMenu();
493 }
494
495 static void autoLoanApplication() {
496     printHeading("AUTO LOAN");
497     notFound();
498     newLoanMenu();
499 }
500
501 static void mortgageLoanApplication() {
502     int id = getUserID();
503     int buildingArea;
504     int creditFacilities;
505     int downPayment;
506     int downPaymentPercentage = 10;
507     int tenor;
508     int debt;
509     int salary = Integer.parseInt(profile[id][3]);
510     double installment;
511     double interest = 7.25;
512     double creditLimit;
513     double debtMax;
514     double installmentMin = salary < 5_000_000 ? salary * 0.5 : salary * 0.55;
515     double salaryMin;
516     printHeading("MORTGAGE LOAN");
517     writeln("""
518         Purpose of Credit
519         1. Buying a house
520         2. Renovating""");
521     String menu = promptedTextInput("menu: ");
522     if (menu.equals("1")) {
523         writeln("""
524             Collateral Type
525             1. House
526             2. Apartment
527             3. Shop""");
528         String collateralType = promptedTextInput("menu: ");
529         if (collateralType.equals("1") || collateralType.equals("2")) {

```



```

530         write("Building Area (m2): ");
531         buildingArea = input.nextInt();
532         if (buildingArea > 70)
533             downPaymentPercentage += 5;
534     }
535     } else if (menu.equals("2")) {
536         downPaymentPercentage += 20;
537     }
538     writeln("How many Credit Facilities do you have");
539     boolean repeat = true;
540     do {
541         write("Credit Facility: ");
542         creditFacilities = input.nextInt();
543         if (creditFacilities < 1) {
544             writeln("Please enter a positive value!");
545         } else {
546             repeat = false;
547         }
548     } while (repeat);
549     if (creditFacilities > 2)
550         downPaymentPercentage += 10;
551     writeln("Maximum 20 years tenor");
552     repeat = true;
553     do {
554         write("Tenor: ");
555         tenor = input.nextInt();
556         if (tenor < 1 || tenor > 20) {
557             writeln("Please enter a value between 1 to 20");
558         } else {
559             repeat = false;
560         }
561     } while (repeat);
562     double interestInMonth = ((interest / 100) / 12);
563     double interestPowerBase = (1 + ((interest / 100) / 12));
564     int tenorMonth = tenor * 12;
565     double inverseReturnOfPoweredInterest = 1 - (1 / powerDouble(interestPowerBase,
↪ tenorMonth));
566     repeat = true;
567     do {
568         creditLimit = installmentMin
569             * (inverseReturnOfPoweredInterest / interestInMonth);
570         debtMax = creditLimit * (1 / ((double) (100 - downPaymentPercentage) / 100));
571         write(String.format("Maximum proposed debt: %,.0f\n", debtMax));
572         write("House price: ");
573         debt = input.nextInt();
574         if (debt > debtMax) {
575             writeln("Please enter a value smaller than the maximum");
576         } else {
577             repeat = false;
578         }
579     } while (repeat);
580     write(String.format("Minimum down payment amount: %,.0f\n", (double) (debt / 100)
↪ * downPaymentPercentage));
581     repeat = true;

```

```

582     do {
583         write("Down payment: ");
584         downPayment = input.nextInt();
585         if (downPayment < (debt / 100) * downPaymentPercentage) {
586             writeln("Please enter a value bigger than the minimum!");
587         } else {
588             repeat = false;
589         }
590     } while (repeat);
591     // region to be refactored as a function
592     if (downPayment > (double) (debt / 100) * downPaymentPercentage) {
593         int[] out = debtReadjustment(installmentMin, inverseReturnOfPoweredInterest,
↪ interestInMonth, downPayment, downPaymentPercentage);
594         debt = out[0];
595         downPayment = out[1];
596     }
597     // endregion
598     double debtInterest = (debt - downPayment) * interestInMonth;
599
600     installment = debtInterest / inverseReturnOfPoweredInterest;
601     salaryMin = installment < 2_500_000 ? installment * 2 : installment * (1 / 0.55);
602     write(String.format("%14s IDR %,d\n", "Installment", (long) installment));
603     write(String.format("%14s IDR %,d\n", "Debt principal", (debt - downPayment)));
604     write(String.format("%14s IDR %,d\n", "Minimum Income", (long) salaryMin));
605
606     String prompt = "Mortgage Application";
607     String ordinal;
608     switch
↪ (String.valueOf(creditFacilities).charAt(String.valueOf(creditFacilities).length() -
↪ 1)) {
609         case '1' -> ordinal = "st";
610         case '2' -> ordinal = "nd";
611         case '3' -> ordinal = "rd";
612         default -> ordinal = "th";
613     }
614     double downPaymentPercentageByDebt = ((double) downPayment / debt) * 100;
615
616     String[] varValue = {
617         String.format(": %d%s", creditFacilities, ordinal),
618         String.format(": IDR %,d", debt),
619         String.format(": IDR %,d", downPayment),
620         String.format(": IDR %,d", (debt - downPayment)),
621         String.format(": %d", tenor),
622         String.format(": %.2f%s", interest, "%"),
623         String.format(": IDR %,d", installment),
624         String.format(": IDR %,d", salaryMin)
625     };
626
627     String[] varName = {
628
629         "Credit facility ",
630         "House price ",
631         String.format("Down payment %.2f%s ", downPaymentPercentageByDebt, "%"),
632         "Debt principal ",

```

```

633         "Tenor ",
634         "Interest ",
635         "installment ",
636         "Minimum income "
637     };
638
639     String barTop =
640 ↪ "=====";
641     String barBot =
642 ↪ "||=====||";
643     String side = "||";
644     int heading = barTop.length();
645     int headingSpacing = (heading - (2 * side.length()) - prompt.length()) / 2;
646     String contentSpacing = String.format("%s%" + (heading - (2 * side.length())) +
647 ↪ "%s%", side, " ", side);
648     String title = String.format("%s%" + headingSpacing + "%s%" + headingSpacing +
649 ↪ "%s%", side, " ", prompt, " ",
650     side);
651     int fit = 0;
652     for (String varNameElement : varName) {
653         if (fit < varNameElement.length()) {
654             fit = varNameElement.length();
655         }
656     }
657     insertMortgageDetail(barTop);
658     insertMortgageDetail(title);
659     insertMortgageDetail(barBot);
660     insertMortgageDetail(contentSpacing);
661     for (int i = 0; i < varName.length; i++) {
662         int paddingLeft = ((heading / 2) - side.length() - fit);
663         int paddingRight = ((heading / 2) - side.length() - varValue[i].length());
664         String content = String.format("%s%" + paddingLeft + "%s%" + fit + "%s%" +
665 ↪ paddingRight + "%s%", side,
666         " ", varName[i], varValue[i], " ", side);
667         insertMortgageDetail(content);
668         insertMortgageDetail(contentSpacing);
669     }
670     insertMortgageDetail(barBot);
671
672     if (confirm()) {
673         creditMortgage[id][0] = debt;
674         creditMortgage[id][1] = tenor;
675         creditMortgage[id][2] = downPayment;
676         creditMortgage[id][3] = installment;
677     }
678     newLoanMenu();
679 }
680
681 static void refinancingLoanApplication() {
682     printHeading("REFINANCING LOAN");
683     notFound();
684     newLoanMenu();
685 }

```

```

682 // endregion
683 static void accountLoanInfo() {
684     printHeading("LOAN STATUS");
685     if (creditMortgage[getUserID()][0] == 0) {
686         notFound();
687     } else {
688         write(creditMortgageDetail[getUserID()]);
689     }
690     write("Exit?");
691     if (confirm()) {
692         loanMenu();
693     } else {
694         accountLoanInfo();
695     }
696 }
697
698 // endregion
699 static void accountInfoMenu() {
700     int id = getUserID();
701     if (profile[id][0] != null) {
702         String content;
703         String prompt = "ACCOUNT INFO";
704         String barTop =
↪ "=====";
705         String barBot =
↪ "||=====||";
706         String side = "||";
707         int heading = barTop.length();
708         String contentSpacing = String.format("%s%" + (heading - (2 * side.length())))
↪ + "s%s", side, " ", side);
709         int titlePadding = (heading - (2 * side.length()) - prompt.length()) / 2;
710
711         String[] varName = {
712             "Name ",
713             "Phone number ",
714             "ID card number ",
715             "salary "
716         };
717         String title = String.format("%s%" + titlePadding + "s%s%" + titlePadding +
↪ "s%s", side, " ", prompt, " ",
718             side);
719
720         writeln(barTop);
721         writeln(title);
722         writeln(barBot);
723         writeln(contentSpacing);
724
725         int fitVarName = 0;
726         for (String varNameElement : varName) {
727             if (fitVarName < varNameElement.length()) {
728                 fitVarName = varNameElement.length();
729             }
730         }
731         int fitVarValue = 0;

```

```

732         for (String varValueElement : profile[id]) {
733             if (fitVarValue < varValueElement.length()) {
734                 fitVarValue = varValueElement.length();
735             }
736         }
737
738         for (int i = 0; i < varName.length; i++) {
739             String var = String.format("%-" + fitVarName + "s: %" + fitVarValue +
↵ "s", varName[i], profile[id][i]);
740             int padding = (heading - (side.length() * 2)
741 ↵ - (var.length() % 2 == 0 ? var.length() : (var.length() + 1))) /
↵ 2;
742             if (var.length() % 2 != 0) {
743                 content = String.format("%s%" + padding + "s%s%" + padding + "s %s",
↵ side, " ", var, " ", side);
744             } else {
745                 content = String.format("%s%" + padding + "s%s%" + padding + "s%s",
↵ side, " ", var, " ", side);
746             }
747             writeln(content);
748             writeln(contentSpacing);
749         }
750         writeln(barBot);
751     } else {
752         printHeading("ACCOUNT INFO");
753         notFound();
754     }
755     write("Exit?\n");
756     if (confirm()) {
757         mainMenu();
758     } else {
759         accountInfoMenu();
760     }
761 }
762
763 static void quitMenu() {
764     printHeading("QUIT SUCCESSFULLY");
765 }
766 // endregion
767 // endregion
768 }

```