Functional Specification

**Project Symphony Drop 2:**

**Member Management Module**

**Version 0.2**

# Revision Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version No. | Date | Revision Description | Author(s) | Approver(s) |
| 0.1 | 23/05/2019 | Initial document based on URS listed below:   * UR for Project Symphony\_Member Movement\_CMX\_Nov 30.docx * Addendum for Member Movement URS\_V1 00.docx | Nurul Iman |  |
| 0.2 | 18/9/2019 | Updated document per comments from FSD walkthrough with business users and technical teams (ESB & G400). | Nurul Iman |  |

# Approval / Signoff

|  |  |  |  |
| --- | --- | --- | --- |
| Approver name and title | Function or Department | Signature or Email attachment | Signoff date |
| Ernie Hee | HOD/ AHS |  |  |
| Jenny Lee | HOD/ CSD |  |  |
| Choo Siew Wan | HOD/ CMX |  |  |
| Sandeep Nottath Nair | CS Strategy/ CSD |  |  |
| Tan Kean Hong | IT Lead - Front End |  |  |
| Wong Jian Ming | IT Lead - Back End |  |  |

# Distribution List

|  |  |
| --- | --- |
| **Company / Department** | **Person / Group** |
| User Lead/ CX Customer Solutions | Eng Mei Keen |
| User Team Manager/ CCU | Tan Hwei Jen |
| User Team Manager/ CS-CSPPM | Fong Yin Cheng |
| User Team Manager/ CMX | Wong Wan Siong |
| User Team Manager/ CMX | Chan Lai Foong |
| User Team Member/ AHS | Zainatul Ashikin Kamal Sabri |
| User Team Member/ AHS | Janet Yeoh |
| User Team Member/ CMX - MiCare | Muthamah Maddacanno |
| IT Lead – Front End | Tan Kean Hong |
| IT Lead – Back End | Wong Jian Ming |
| BA – Back End | Madhan Kumar |
| IT SME – G400 | Wong Jian Ming |
| IT SME – ESB | Sharon Diana Saldanha |
| IT SME – CPF | Vijay Chandra Jaligama |
| IT SME – Backend Services | Srinivasa Rao Pamidimukkala |
| IT SME – Bicor | Chang Chee Wun / T.Komalavalli / Wong Choon Keat |
| IT SME – Email Gateway | Chang Chee Wun / Nik Fariz Nik Mohamed Noor |
| IT SME – Case360 | Chang Chee Wun / Nik Fariz Nik Mohamed Noor / Mohammad Azlan Yusof |

**Table of Contents**

[Revision Summary 2](#_Toc19727649)

[Approval / Signoff 3](#_Toc19727650)

[Distribution List 4](#_Toc19727651)

[1 Objective 7](#_Toc19727652)

[2 Scope 7](#_Toc19727653)

[2.1 In Scope 7](#_Toc19727654)

[2.2 Out Scope 7](#_Toc19727655)

[3 Current processing 7](#_Toc19727656)

[4 Business Flow Diagram 7](#_Toc19727657)

[5 Assumptions, dependencies and constraints 8](#_Toc19727658)

[5.1 Assumptions 8](#_Toc19727659)

[5.2 Dependencies 8](#_Toc19727660)

[5.3 Constraints 8](#_Toc19727661)

[6 Requirement trace index 8](#_Toc19727662)

[7 Detailed functional descriptions 9](#_Toc19727663)

[7.1 FR-001: Change Member Details 9](#_Toc19727664)

[7.1.1 Change Member Details 10](#_Toc19727665)

[7.2 FR-002: Change Dependent Details 12](#_Toc19727666)

[7.2.1 Change Dependent Details 12](#_Toc19727667)

[7.3 FR-003: Change Plan and Salary 14](#_Toc19727668)

[7.3.1 Change Plan and Salary 14](#_Toc19727669)

[7.4 FR-004: Change Restricted Panel 16](#_Toc19727670)

[7.4.1 Change Restricted Panel 16](#_Toc19727671)

[7.5 FR-005: Request Medical Card 18](#_Toc19727672)

[7.5.1 Request Medical Card 18](#_Toc19727673)

[7.6 FR-006: Bulk Member Termination 19](#_Toc19727674)

[7.6.1 Bulk Member Termination 19](#_Toc19727675)

[8 Data requirements 20](#_Toc19727676)

[9 Other requirements 20](#_Toc19727677)

[10 Testing consideration 20](#_Toc19727678)

[11 Interface consideration 20](#_Toc19727679)

[12 Conversion consideration 21](#_Toc19727680)

[12.1 Policy Type Criteria 21](#_Toc19727681)

[12.2 Policy Status Criteria 21](#_Toc19727682)

[12.3 Member Status Criteria 22](#_Toc19727683)

[13 Reference 23](#_Toc19727684)

[13.1 Reference Documents 23](#_Toc19727685)

[13.2 Systems Reference 23](#_Toc19727686)

[13.3 Acronyms 23](#_Toc19727687)

[14 Appendix 24](#_Toc19727688)

[14.1 Wireframes 24](#_Toc19727689)

[14.1.1 Change Member Details Wireframes 24](#_Toc19727690)

[14.1.2 Change Dependent Details Wireframes 26](#_Toc19727691)

[14.1.3 Change Plan and Salary Wireframes 29](#_Toc19727692)

[14.1.4 Change Restricted Panel Wireframes 32](#_Toc19727693)

[14.1.5 Request Medical Card Wireframes 33](#_Toc19727694)

[14.1.6 Bulk Member Termination Wireframes 34](#_Toc19727695)

# Objective

This document outlines the functional specifications of the intermediate systems for Project Symphony Drop 2 Member Management on the My AIA Corporate portal.

My AIA Corporate portal is a self-service portal that allows user to perform employee benefit related task without having to contact the call center for information.

This document does not include any details of processes, validations or data mapping that happens on portal and source systems. The details on portal and source system are covered in separate documents.

# Scope

## In Scope

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Sub-Modules | Features | Priority |
| 1 | Edit Member Details | * Member personal info * Member bank info | High |
| 2 | Edit Dependent Details | * Dependent personal info | Medium |
| 3 | Update Restricted Panel | * Update restricted panel for member and dependent | Low |
| 4 | Change Plan | * Change plan for member and dependent | High |
| 5 | Medical Card Request | * Request physical medical card | Low |
| 6 | Bulk Member Termination | * Terminate member in bulk | High |
| 7 | Bulk Dependent Termination | * Terminate dependent in bulk | Medium |

## Out Scope

|  |  |  |
| --- | --- | --- |
| No. | Sub-Modules | Remarks |
| 1 | Transfer Member | Transfer member from one or multiple policy to another policy is descoped. |
| 2 | Bulk Upload | Bulk upload on member transaction is descoped. |

# Current processing

Current process is handle via AMSC but it is not real-time. Since this is a new portal and real-time, hence new APIs will be created.

# Business Flow Diagram

The flow diagrams are separated by sections and are shown in each section before the functional descriptions.

# Assumptions, dependencies and constraints

## Assumptions

|  |  |
| --- | --- |
| **No.** | **Assumptions Description** |
| 1 |  |

## Dependencies

|  |  |  |
| --- | --- | --- |
| **No.** | **Dependencies Description** | **PIC/System** |
| 1 |  |  |

## Constraints

|  |  |
| --- | --- |
| **No.** | **Constraints Description** |
| 1 |  |

# Requirement trace index

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Business requirement (BR) number | Functional requirement (FR) number | Description | Online change | Batch change | Screen design | New process flow |
| 4.1 - Requirements for G400 enhancement – Member Movement | - | * This is covered in G400 document. | Yes | No | Yes | Yes |
| 4.2 - Requirements for Portal G400 – Member Movement | FR-001,  FR-002,  FR-003,  FR-004,  FR-005,  FR-006 | * Edit Member/ Dependent Details * Change Plan * Update Restricted Panel * Medical Card Request * Bulk Member/Dependent Termination | Yes | No | Yes | Yes |

# Detailed functional descriptions

## FR-001: Change Member Details





|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Change Member Details  * User can change the member details in a single submission, including the below:   + Employment details.   + Plan details.   + Dependent details.   + Restricted panel, if new plan is selected. * Refer to the wireframes [here](#_Change_Member_Details). | 1. User can change the member details, including employment details, plan details, dependents’ details and restricted panel (if new plan is selected) in a single submission. 2. After user has edit the member details and employee details, and click on Proceed button, user will call *ValidateChangeMember* API. 3. Once the *ValidateChangeMember* API is called, ESB will:    * Validate the changed member details and employee details per rules below:      1. Check if DOB changed, then check for eligible policy based on Renew Up to Age.    * If validation failed, then pass error message to portal.    * If validation passed, then pass the success validation result to portal to allow user to change plan. 4. Once user clicks on Get Plan button, portal will call *RetrieveEligiblePlanAndCoverage* API to get the list of eligible plans based on the new effective date (submission date) for member. 5. Once the *RetrieveEligiblePlanAndCoverage* API is called, ESB will:    * Retrieve all the eligible plans for the member from G400 Datamart based on the new effective date.    * If failed to retrieve, then pass error message to portal.    * If no plan found, then pass no eligible plan to portal.    * If plan found, then pass the eligible plans to portal. 6. If there is eligible plan for member, user need to select a new plan, before proceeding. Once user clicks on Proceed button, if the member has dependents, then user can change the dependents’ plan as well. 7. To change the member salary, user can directly change if the plan allows for salary, and then proceed. User can also update the bank account details if the plan allows it. The bank account can be blanked out if previously has value. 8. To change the dependents’ plan, user needs to click on the Edit button and then clicks on Get Plan button to continue with the change journey. 9. Once user clicks on Get Plan button, portal will call *RetrieveEligiblePlanAndCoverage* API to get the list of eligible plans based on the new effective date for dependents. 10. Once the *RetrieveEligiblePlanAndCoverage* API is called, ESB will:     * Retrieve all the eligible plans for the dependents from G400 Datamart based on the new effective date.     * If failed to retrieve, then pass error message to portal.     * If no plan found, then pass no eligible plan to portal.     * If plan found, then pass the eligible plans to portal. 11. If there is eligible plan for dependents, user need to select a new plan, before proceeding. 12. Once user clicks on Proceed button, if the new member/dependents plan requires restricted panel, then user can change the restricted panel for member and dependents as well. 13. To change the member and dependents’ restricted panel, the flow will be the same as in the [Change Restricted Panel](#_Change_Restricted_Panel_1) section (point 2 till 8). 14. On the change summary page before user submits, portal will call *CalculateSumAssured* API to get the calculated sum assured. 15. Once user clicks on the Submit button, portal will call *SubmitChangeMember* API to update the member and dependents (if exist) changes. 16. Once the *SubmitChangeMember* API is called, ESB will:     * Break down each member and dependents’ policy into separate transaction record. 1 member’s policy is considered as 1 transaction record and 1 dependent’s policy is considered as another transaction record.     * Generate the transaction reference no.     * Add the transaction record with transaction reference in Staging DB.     * Check if there is action = Terminate for any of the member or dependent’s policy. If there is, then update the details first before terminating in G400 using the submission date as termination effective date.     * If there is any Claim Incurred Date for the member or his/her family, then the termination effective date will be 1 day after that.     * If there is no termination, then update all the changes for the member and dependents to G400 as per selected.     * If Bank Account number is blanked out (previously there is value), then terminate the bank account with the effective date submitted.     * If member name or member ID is changed, then update the details in G400 first before request to issue new medical card.     * If member DOB is changed, then check the premium value if it’s different. If it’s different, then set transaction status as Manual.     * Set the transaction status as Manual (Pending Verification) as it requires user intervention for scenarios below:       1. Member found in L4.       2. Member gender changed and MAT product exist.     * Update the transaction status for each transaction record based on the response from G400. For Nationality=MAL, based on G400 response, set transaction status as Manual (Pending Verification). | ESB,  SP,  Datamart,  G400 |

## FR-002: Change Dependent Details

|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Change Dependent Details  * User can change the dependent details including the below:   + Plan details.   + Restricted panel, if new plan is selected. * Refer to the wireframes [here](#_Change_Dependent_Details). | 1. User can change the dependent’s details, including plan and restricted panel (if new plan is selected) in a single submission. 2. User needs to select the dependent which to be changed, then clicks on Proceed button to continue with the change journey. 3. User needs to click on the Edit button below each dependent section (details, plans) to change the dependent details and plans respectively. 4. After user has updates the dependent details and click on Verify button, user will call *ValidateChangeDependant* API. 5. Once the *ValidateChangeDependant* API is called, ESB will:    * Validate the changed dependent details per rules below:      1. Check if DOB changed, then check for eligible policy based on Renew Up to Age.    * If validation failed, then pass error message to portal.    * If validation passed, then pass the success validation result to portal to allow user to change plan. 6. Once user clicks on Get Plan button, portal will call *RetrieveEligiblePlanAndCoverage* API to get the list of eligible plans based on the new effective date (submission date) for dependent. 7. Once the *RetrieveEligiblePlanAndCoverage* API is called, ESB will:    * Retrieve all the eligible plans for the dependent from G400 Datamart based on the new effective date.    * If failed to retrieve, then pass error message to portal.    * If no plan found, then pass no eligible plan to portal.    * If plan found, then pass the eligible plans to portal. 8. If there is eligible plan for dependent, user need to select a new plan, before proceeding. Once user clicks on Proceed button, if the new plan requires restricted panel, then user can change the restricted panel as well. 9. To change the restricted panel, the flow will be the same as in the [Change Restricted Panel](#_Change_Restricted_Panel_1) section (point 2 till 8). 10. On the change summary page before user submits, portal will call *CalculateSumAssured* API to get the calculated sum assured. 11. Once user clicks on the Submit button, portal will call *SubmitChangeDependant* API to update the dependents changes. 12. Once the *SubmitChangeDependant* API is called, ESB will:     * Break down each dependents’ policy into separate transaction record. 1 dependent’s policy is considered as another transaction record.     * Dependent transaction records are all independent of each other without sequence.     * Generate the transaction reference no.     * Add the transaction record with transaction reference in Staging DB.     * Check if there is action = Terminate for any of the dependent’s policy. If there is, then update the details first before terminating in G400 using the submission date as termination effective date.     * If there is any Claim Incurred Date for the dependent, then the termination effective date will be 1 day after that.     * If there is gender changed for spouse and MAT product exist, then set transaction status as Manual as this require user to check.     * If the dependent’s ID number is removed, then update as blank to G400.     * Apart from the above-mentioned cases, update all the changes for the dependent to G400 as per selected.     * Update the transaction status for each transaction record based on the response from G400. | ESB,  SP,  Datamart,  G400 |

## FR-003: Change Plan and Salary



|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Change Plan and Salary  * User can change the member’s plan and/or salary, including the below:   + Dependent plan.   + Restricted panel, if new plan is selected. * Refer to the wireframes [here](#_Change_Plan_Wireframes). | 1. User can change the member’s plan and/or salary, including change the dependents’ plan and change member/dependent restricted panel (if new plan is selected) in a single submission. 2. User needs to select the policy with a new effective date to change the plan and/or salary, then clicks on Proceed and then Edit buttons to continue with the change journey. 3. Once user clicks on Get Plan button, portal will call *RetrieveEligiblePlanAndCoverage* API to get the list of eligible plans based on the new effective date for member. 4. Once the *RetrieveEligiblePlanAndCoverage* API is called, ESB will:    * Retrieve all the eligible plans for the member from G400 Datamart based on the new effective date.    * If failed to retrieve, then pass error message to portal.    * If no plan found, then pass no eligible plan to portal.    * If plan found, then pass the eligible plans to portal. 5. If there is eligible plan for member, user need to select a new plan, before proceeding. 6. To change salary, user can directly change if the plan allows for salary, and then proceed. User can also update the bank account details if the plan allows it. 7. Once user clicks on Proceed button, if the member has dependents, then user can change the dependents’ plan as well. 8. To change the dependents’ plan, user needs to click the Edit button and then clicks on Get Plan button to continue with the change journey. 9. Once user clicks on Get Plan button, portal will call *RetrieveEligiblePlanAndCoverage* API to get the list of eligible plans based on the new effective date for dependents. 10. Once the *RetrieveEligiblePlanAndCoverage* API is called, ESB will:     * Retrieve all the eligible plans for the dependents from G400 Datamart based on the new effective date.     * If failed to retrieve, then pass error message to portal.     * If no plan found, then pass no eligible plan to portal.     * If plan found, then pass the eligible plans to portal. 11. If there is eligible plan for dependents, user need to select a new plan, before proceeding. 12. Once user clicks on Proceed button, if the new member/dependents plan requires restricted panel, then user can change the restricted panel for member and dependents as well. 13. To change the member and dependents’ restricted panel, the flow will be the same as in the [Change Restricted Panel](#_Change_Restricted_Panel_1) section (point 2 till 8). 14. On the change summary page before user submits, portal will call *CalculateSumAssured* API to get the calculated sum assured. 15. Once user clicks on the Submit button, portal will call *SubmitChangeMember* API to update the member and dependents (if exist) changes. 16. Once the *SubmitChangeMember* API is called, ESB will:     * Break down each member and dependents’ policy into separate transaction record. 1 member’s policy is considered as 1 transaction record and 1 dependent’s policy is considered as another transaction record.     * Each member transaction record must be completed first before BO for dependent.     * Generate the transaction reference no.     * Add the transaction record with transaction reference in Staging DB.     * Check if there is action = Terminate for any of the member or dependent’s policy. If there is, then update the details first before terminating in G400 using the submission date as termination effective date.     * If there is any Claim Incurred Date for the member or his/her family, then the termination effective date will be 1 day after that.     * If underwriting flag = 6, then set transaction status as Manual.     * If there is no termination, then update all the changes for the member and dependents to G400 as per selected.     * Update the transaction status for each transaction record based on the response from G400. | ESB,  SP,  Datamart,  G400 |

## FR-004: Change Restricted Panel



|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Change Restricted Panel  * User can change the restricted panel for member or dependent or both. * Refer to the wireframes [here](#_Change_Restricted_Panel). | 1. User can change the restricted panel for member or dependent or both in a single submission. This change can be:    * Add new panel.    * Change existing panel to a different panel.    * Remove existing panel. 2. When user clicks on the Change GP Panel option on View Member page, portal will call *RetrievePanelAddress* API to get the full address of the existing panel. 3. Once the *RetrievePanelAddress* API is called, ESB will:    * Retrieve the full address of all the existing panels for member and dependent from G400 Datamart and Medi-connect DB based on the provider no. including:      1. Town      2. Postcode      3. State    * If failed to retrieve, then pass error message to portal.    * If no data found, then pass no record found to portal.    * If data found, then pass the address to portal to be displayed. 4. To add or change existing panel, user needs to search the restricted panel and selects the new panel from the result. 5. Once user enters the search criteria and clicks on Search button on the restricted panel search page, portal will call *RetrieveRestrictedPanelList* API to retrieve the restricted panel list. 6. Once the *RetrieveRestrictedPanelList* API is called, ESB will:    * Retrieve the list of panels from G400 Datamart and Medi-connect DB based on the search criteria and the Restricted Panel Type (Public/Private).    * If failed to retrieve, then pass error message to portal.    * If no data found, then pass no record found to portal.    * If data found, then pass the listing to portal to be displayed. 7. Once user selects the restricted panel for member and dependents and clicks on the Proceed button, portal will call *ValidateRestrictedPrivatePanel* API to validate the selected restricted panel. 8. Once the *ValidateRestrictedPrivatePanel* API is called, ESB will:    * Validate the selected panel for each member and dependents’ policy if it is in the member/dependent plan’s panel network (e.g. INGNET / Petronas NET, etc.).    * If validation failed, then pass error message to portal.    * If validation passed, then pass the success validation result to portal to allow user to submit. 9. Once user clicks on the Submit button, portal will call *SubmitChangePanel* API to update the panel changes. 10. Once the *SubmitChangePanel* API is called, ESB will:     * Break down each member and dependents’ policy into separate transaction record. 1 member’s policy is considered as 1 transaction record and 1 dependent’s policy is considered as another transaction record.     * Generate the transaction reference no.     * Add the transaction record with transaction reference in Staging DB.     * For each transaction record, terminate the old the restricted panel and update the new one (if exist) for the member and dependents to G400 as per selected.     * Update the transaction status for each transaction record based on the response from G400. 11. To remove existing panel, user needs to remove the existing panel and clicks on Proceed button. The same process as described in point (7) until point (10) above will be performed. 12. User not allowed to remove all restricted panel and not add one (at least 1 is mandatory for restricted panel type). | ESB,  SP,  Datamart,  Medi-connect DB,  G400 |

## FR-005: Request Medical Card



|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Request Medical Card  * User can request for a new medical card for member or dependent or both. * User can submit the request only if there is no request submitted within the last 7 days for the selected member policy. * Refer to the wireframes [here](#_Request_Medical_Card). | 1. User can select which member or dependent or both to request medical card in a single submission. 2. Once user selects and clicks on submit button, portal will call *SubmitMedicalCardRequest* API to issue the card. 3. Once theAPI is called, ESB will:    * Retrieve the latest medical card issue date for selected member/dependent from G400 Datamart based on the member client no.    * If failed to retrieve, then pass error message to portal.    * If no date found, then proceed to point (4) below.    * If date is found, then validate if the latest issue date is within 7 days from the submission date (submission date – latest issue date <= 7 days).    * If validation failed, then pass error message to portal to block user from submit.    * If validation passed, then proceed to point (3) below. 4. If the validation passed or no date found, ESB will:    * Break down each member and dependent policy into separate transaction record. 1 member’s policy is considered as 1 transaction record and 1 dependent’s policy is considered as 1 transaction record.    * Generate the transaction reference no.    * Add the transaction record with transaction reference in Staging DB.    * For each transaction record, submit the card request to G400.    * If member transaction record failed, then the remaining transaction will set status to Failed as well.    * G400 will issue a new card and assign a new card number for each transaction record.    * Update the transaction status for each transaction record based on the response from G400. | ESB,  SP,  Datamart,  G400 |

## FR-006: Bulk Member Termination

|  |  |  |
| --- | --- | --- |
| Business Process | Functional Descriptions | Systems |
| Bulk Member Termination  * User can perform member termination by:   + Single or multiple member termination within a single submission.   + Maximum of 50 members can be terminated at a time. * Refer to the wireframes [here](#_Bulk_Member_Termination). | 1. For multiple member termination, user can select which member to terminate up to a maximum of 50 members in a single submission. 2. Once user selects the members to terminate and clicks on Terminate button, portal will call *RetrieveBulkMembers* API to get the necessary info for member termination. 3. Once the *RetrieveBulkMembers* API is called, ESB will:    * Retrieve all the member info from G400 Datamart based on member client no. for each selected member.      1. Member details      2. Member policy details      3. Dependent details    * If failed to retrieve, then pass error message to portal.    * If no data found, then pass no record found to portal.    * If data found, then pass the listing to portal to be displayed. 4. User need to select the termination effective date for each member, before proceeding. Once user clicks on Proceed button, *ValidateTerminateMember* API will be called to validate the members’ termination effective date. 5. Once the *ValidateTerminateMember* API is called, ESB will:    * Retrieve the Claim Incurred Date (where claim status CA, CC or CP) from G400 Datamart for the members and all their dependents (if exist) to validate the member termination date.    * Check the termination date for each member’s policy per below rules:      1. Not backdated more than 30 days.      2. Not earlier than member enrolment effective date.      3. Not later than Policy Insured To Date.      4. Not earlier than the latest Claim Incurred Date for the member and all his/her dependents (if exist) of each policy.      5. Not later than the member’s existing future termination dates, if exist.    * If validation failed, then pass error message with failed reason to portal for each failed member policy.    * If validation passed, then pass the success validation result to portal. 6. If the validation passed, user can proceed to submit the termination. Once user clicks on Terminate button, *SubmitTerminateMember* API will be called to terminate the selected members with their dependents, if exist. 7. Once the *SubmitTerminateMember* API is called, ESB will:    * Break down each member’s policy into separate transaction record. 1 member’s policy is considered as 1 transaction record.    * Generate the transaction reference no.    * Add the transaction record with transaction reference in Staging DB.    * For each transaction record, check if there is any dependent under the member which has future termination date that is earlier than the member selected termination date. If there is, DO NOT update this dependent’s termination date.    * Apart from above case, update the termination date for members and all their dependents with the selected termination date.    * Update the transaction status for each transaction record based on the response from G400. | ESB,  SP,  Datamart,  G400 |

# Data requirements

Data mapping will be in a separate excel file which covers mapping between portal to ESB and ESB to G400 and Case360.

# Other requirements

Not applicable.

# Testing consideration

Test cases will be created by the Business User after the requirements’ freeze period, which will be later than this FSD completion. Hence it will not be included in this document.

However, the test cases are necessary during the technical design to ensure that the development covers all test scenarios.

# Interface consideration

Not applicable.

# Conversion consideration

## Policy Type Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Portal View | G400 Value | ESB Logic |
| 1 | Conventional | Company 3 | Check field CHDRCOY, where Company 3 = Conventional |
| 2 | Takaful | Company 4 | Check field CHDRCOY, where Company 4 = Takaful |

## Policy Status Criteria

|  |  |
| --- | --- |
| **Step 1: ESB Logic** | |
| 1. Exclude Policy Corrupted = Y and Declined policy in G400. | |
| 1. If G400 status = PN, return status as “**New Business In Progress**”. 🡪 excluded from portal 2. If G400 status = PR, return status as “**Renewal In Progress**”. 3. If G400 policy status = PN or PR, Check in G400 TM99V, if the policy found in that table with valid flag = Y, then status is **Suspended**. 4. If G400 status = LA or CA, return status as “**Inactive**”. | |
| 1. If G400 status is IF or PE and Policy Inactive indicator is not “Y”, then check in Case360 using logic below. | |
| **Step 2: Case360 Logic** | |
| **POS** | **Renewal** |
| 1. Check latest Active, Closed/ Completed cases for Non-adjustment with policy termination box checked only. 2. Renewal Billing adjustment user not null only. 3. If found, compare the effective date <= today return status “**Inactive**”. 4. If found, compare the effective date > today return status “**Active**”. | 1. Check latest Active, Closed/ Completed cases and renewal adjustment user not null only. 2. If found and tick lost, Case360 PolicyFrom date <= today, return status “**Expired**”. 3. If found and tick lost, Case360 PolicyFrom date > today, return status “**Active**”. 4. If found and untick lost, Case360 PolicyFrom date > G400 PolicyTo date, return status “**Renewal In Progress**”. 5. If found and untick lost, Case360 PolicyFrom date <= G400 PolicyTo date, then check:    1. If G400 date <= today, then return “**Expired**”    2. If G400 date > today, then return “**Active**” |
| 1. If cases found in both type, take the latest case. 2. If cases NOT found in both type (POS and Renewal), and:    1. If G400 date <= today, then return “**Expired**”    2. If G400 date > today, then return “**Active**” 3. If status is null from Case360, then check:    1. If G400 date <= today, then return “**Expired**”    2. If G400 date > today, then return “**Active**” | |
| **Step 3: ESB Logic** | |
| 1. If policy status is **Active** from case360, Check in G400 TM99V, if the policy found in that table with valid flag = Y, then status is **Suspended**. | |

## Member Status Criteria

|  |  |  |
| --- | --- | --- |
| No. | Status on Portal | Remarks |
| 1 | Active | Member advance termination date is greater than current date. |
| 2 | Pending Acceptance | * Member underwriting decision as “7” = Pending for at least 1 product. * Non-underwriting related (if STP using L400 database, details refer to 3.0 Addendum under (ii)L400 with Remark “Pending for verification”) |
| 3 | Declined | Member underwriting decision as “4” = declined with zero Sum Assured (refer to all product) |
| 4 | Terminated | Member with terminated effective date except advance termination date which is less than current date.   1. Policy status as Lapse (LA) |

Only show from the latest policy:

* For example (latest policy indicated in red):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mbr# | Client# | Pln | Name | Attachment date | Terminate date |
| 00001 00 | 60990575 | 238 | WONG HEE | 01/04/2015 | 01/01/2016 |
| 00005 00 | 60990575 | 238 | WONG HEE | 01/01/2016 | 01/10/2017 |
| 00010 00 | 60990575 | 238 | WONG HEE | 01/11/2017 | 01/05/2018 |

# Reference

## Reference Documents

|  |  |  |
| --- | --- | --- |
| **No.** | **File Name** | **File Description** |
| 1 | User Access Matrix\_V6 10\_20181221.xlsx | User Access Matrix. |
| 2 | UR for Project Symphony\_Member Movement\_CMX\_Nov 30.docx | User Requirement for Project Symphony Member Management. |
| 3 | Addendum for Member Movement URS\_V1 00.docx | Addendum for Member Movement UR. |
| 4 | Member Management FD Feedback v3.xlsx | Member Movement scenarios on portal and ESB. |
| 5 | Symphony Drop 2\_Module & Matrix\_280819.xlsx | Member Movement simple and complex scenarios. |
| 6 | MyAIA Corporate Portal - Integration Data Mapping v6.7.xlsx | Integration Data Mapping between portal and ESB. |

## Systems Reference

|  |  |
| --- | --- |
| **Application** | **Scope of Work** |
| ESB | Average of 80~100 new APIs to be created, for transactions & pulling data |
| Staging Services | Creation of new ones to facilitate real-time transactions between portal & backend |
| CPF | Enhancements to manage account/profile creation & role management |
| G400 | Enhancements to process real-time transactions from Portal |
| G400 MIMIX | Real-time replication of G400 database which can fulfilled the real-time request for MY AIA Corporate |
| Medi-Connect | Enhancement to pulling the claims info for MY AIA Corporate |
| Case 360 | Enhancement to pulling the Policy info for MY AIA Corporate |
| Callidus | Enhancement to pulling the Agent Hierarchy for MY AIA Corporate |
| CRM | Enhancement to pulling the claims info for MY AIA Corporate |
| Bicor/Print Agent | Enhancement to pre-generate all the Claims, Policies and Billing Documents for My AIA Corporate |
| Email Gateway | An Email Gateway allowed the scheduled service to massage the data and send all email notifications |

## Acronyms

|  |  |
| --- | --- |
| **Acronyms** | **Descriptions** |
| ESB | Enterprise Service Bus |
| AMSC | AIA MyService Corporate |
| CPF | Core Portal Framework |
| CRM | Enhancements to process real-time transactions from Portal |
| API | Application Programming Interface |
| AEM | Adobe Experience Manager |
| Portal | MyAIA Corporate Portal |

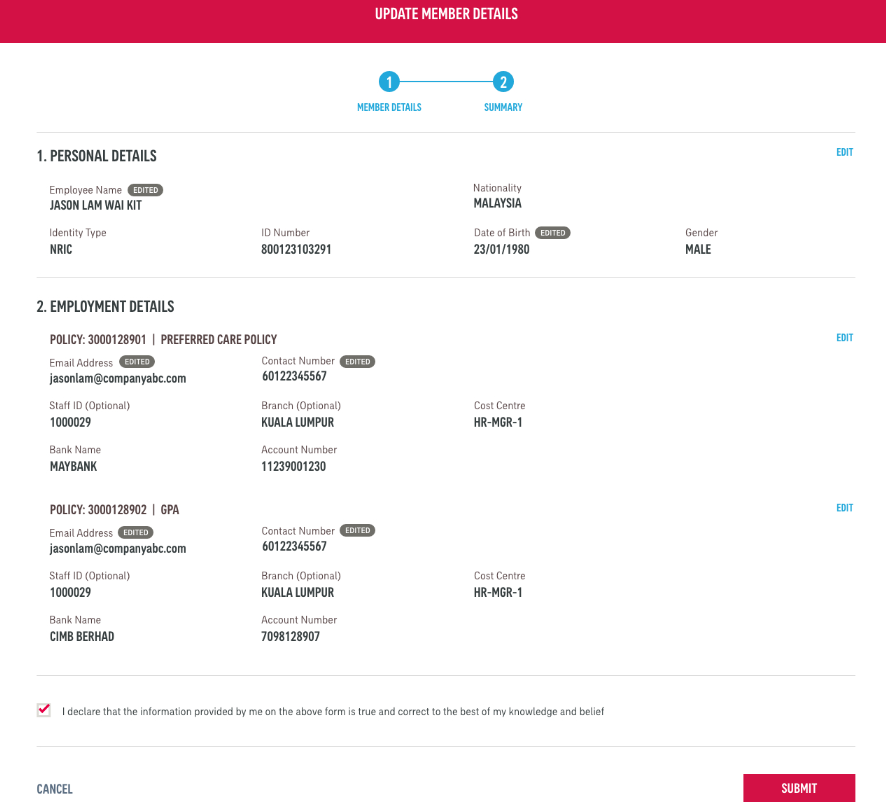
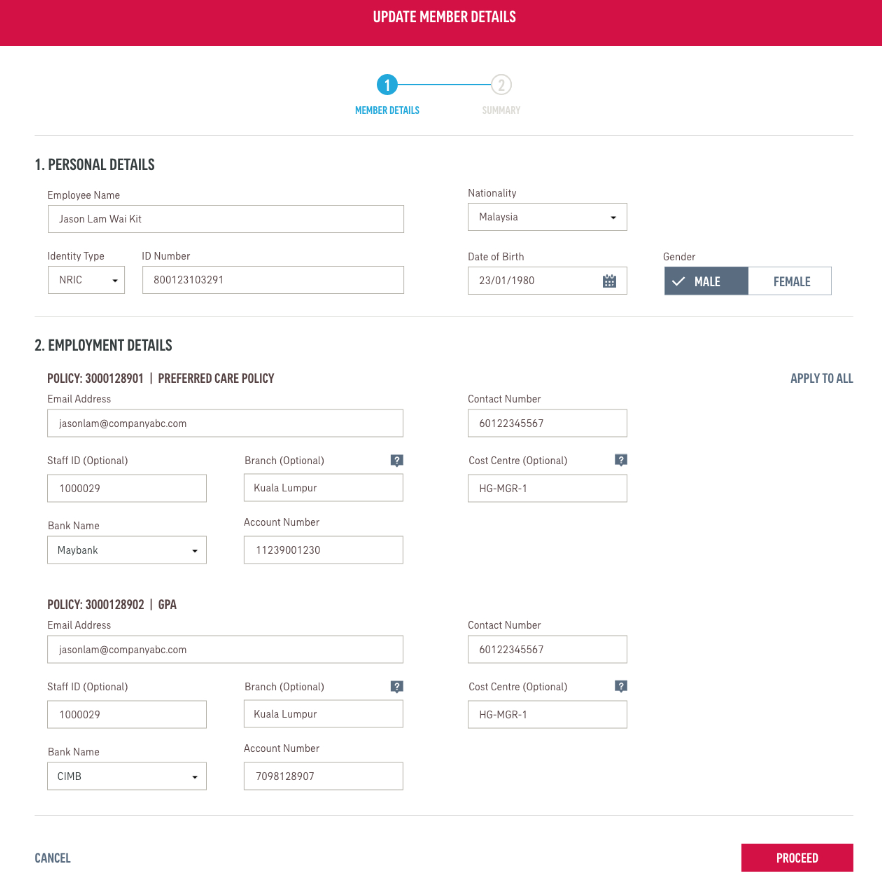
# Appendix

## Wireframes

The wireframes can be accessed via the URL: <https://projects.invisionapp.com/share/3MSQPXN6GKA#/screens/371262849>

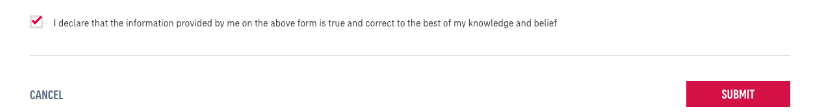
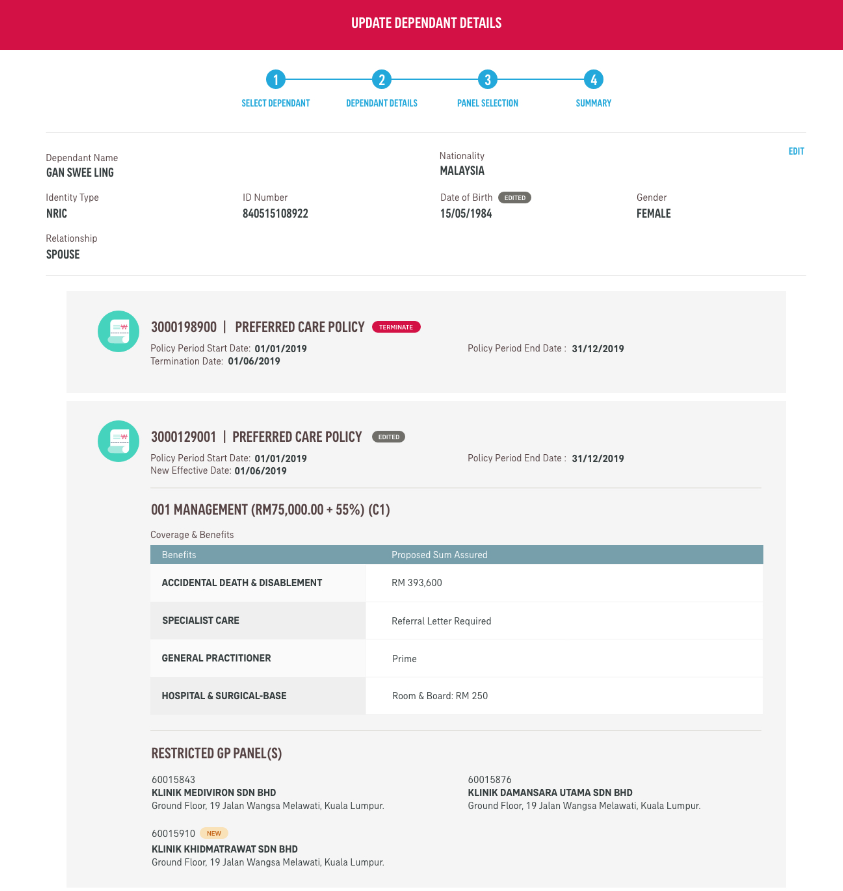
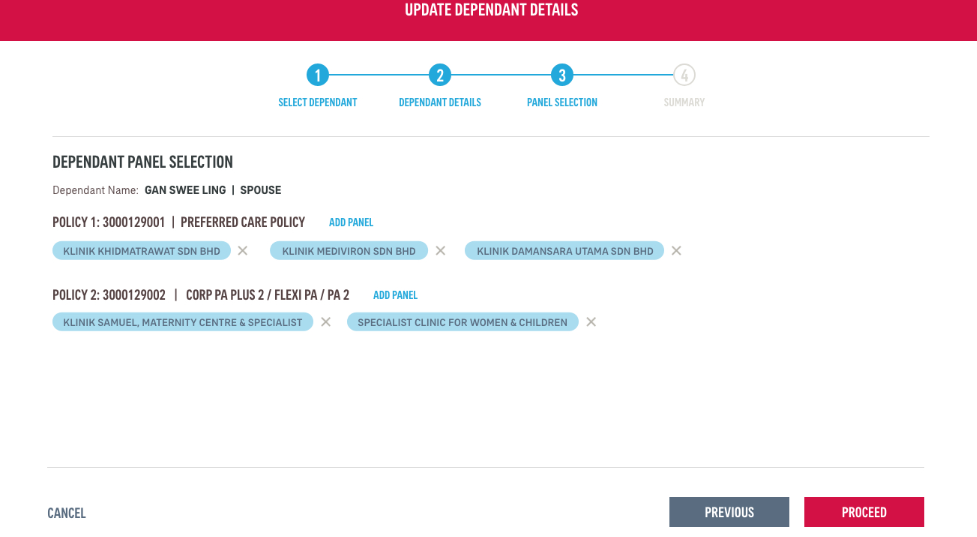
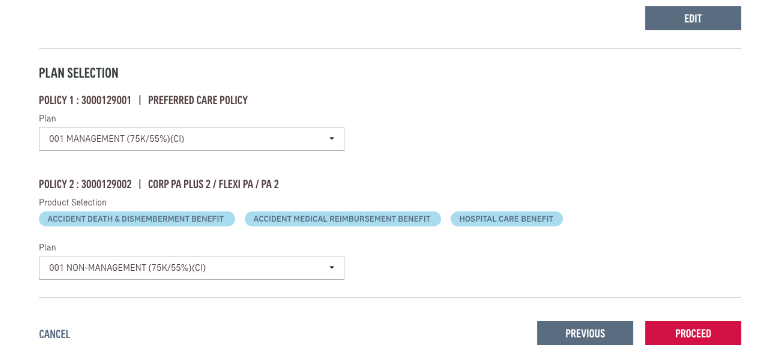
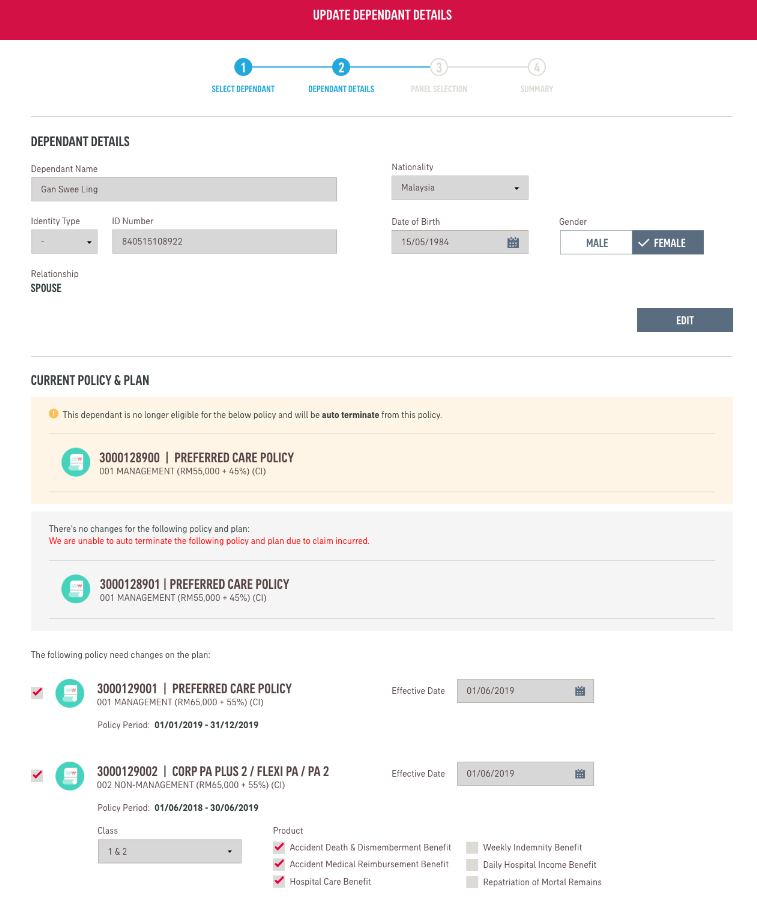
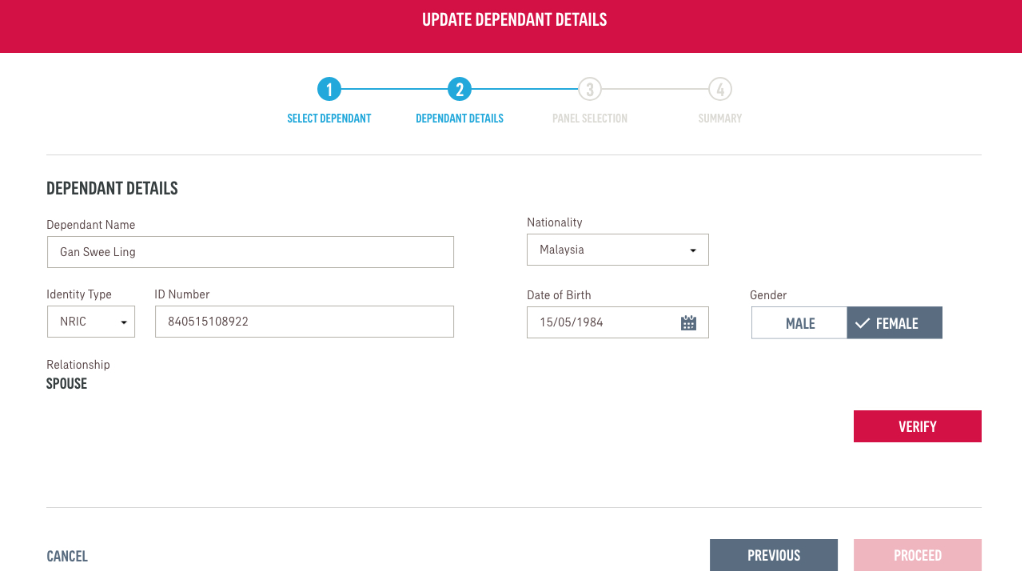
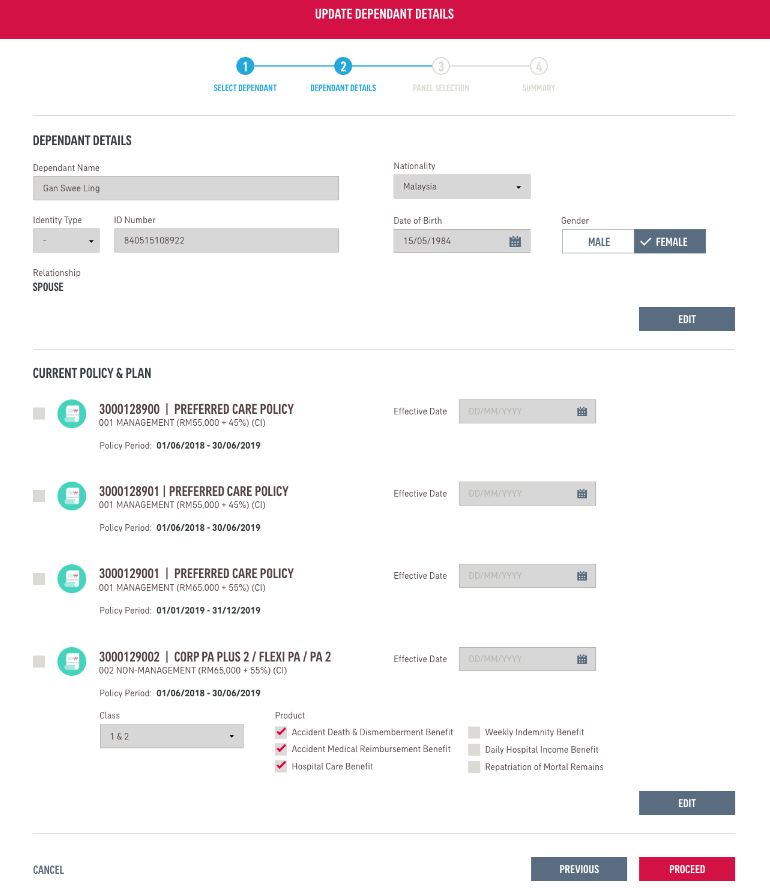
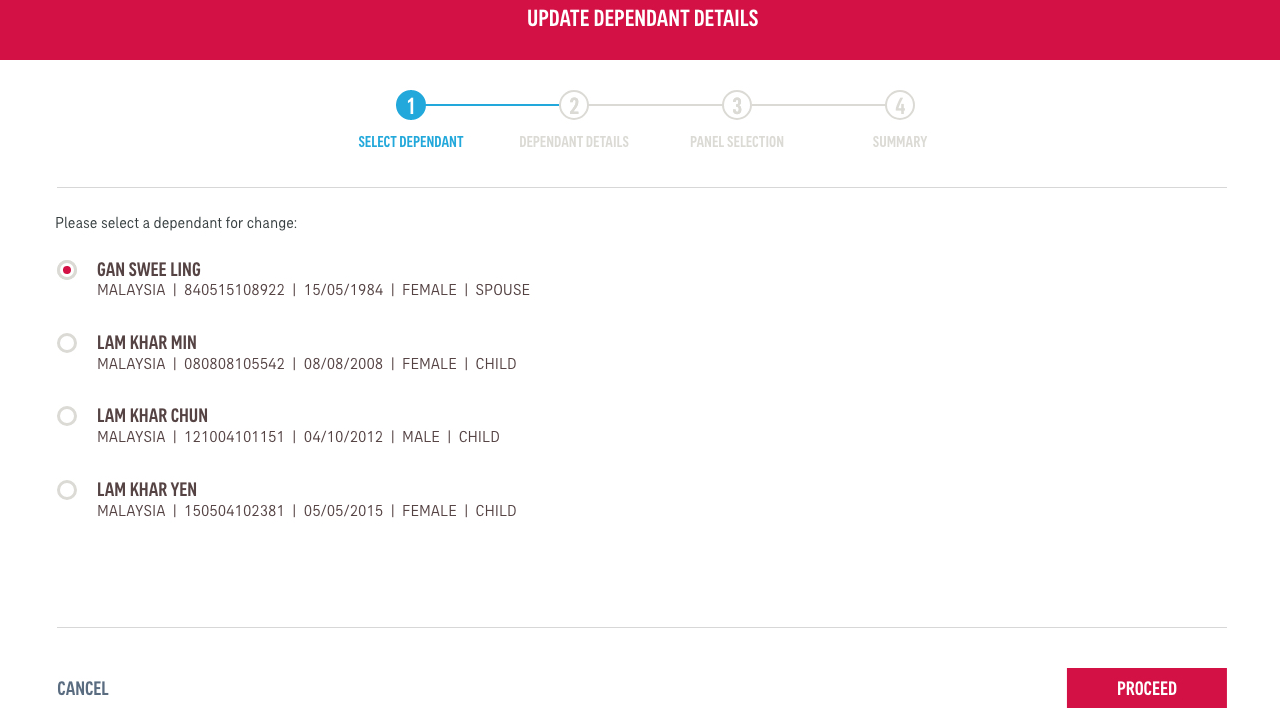
### Change Member Details Wireframes

The change member details wireframes are as below.



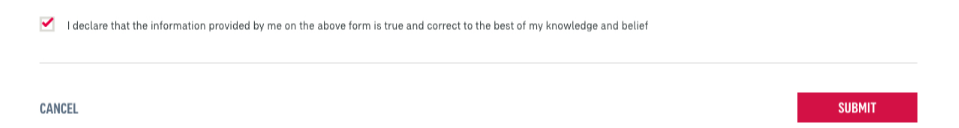
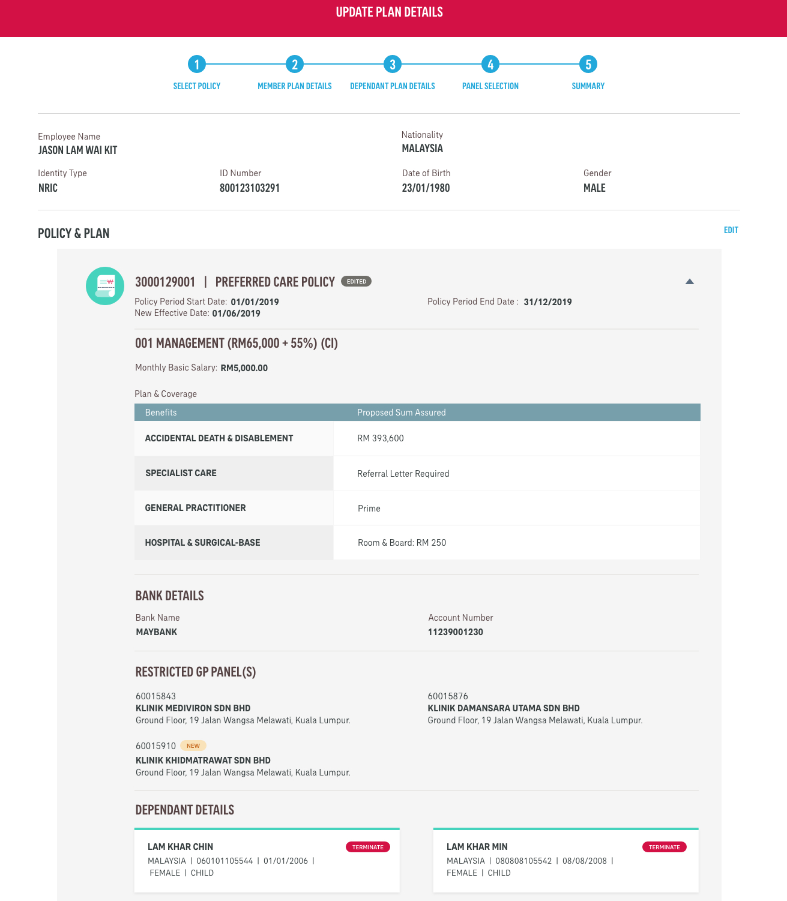
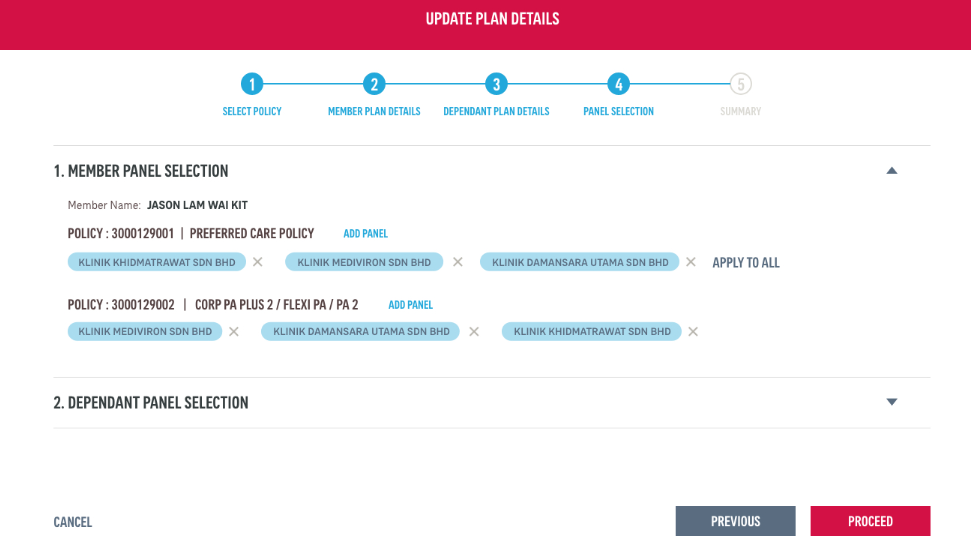
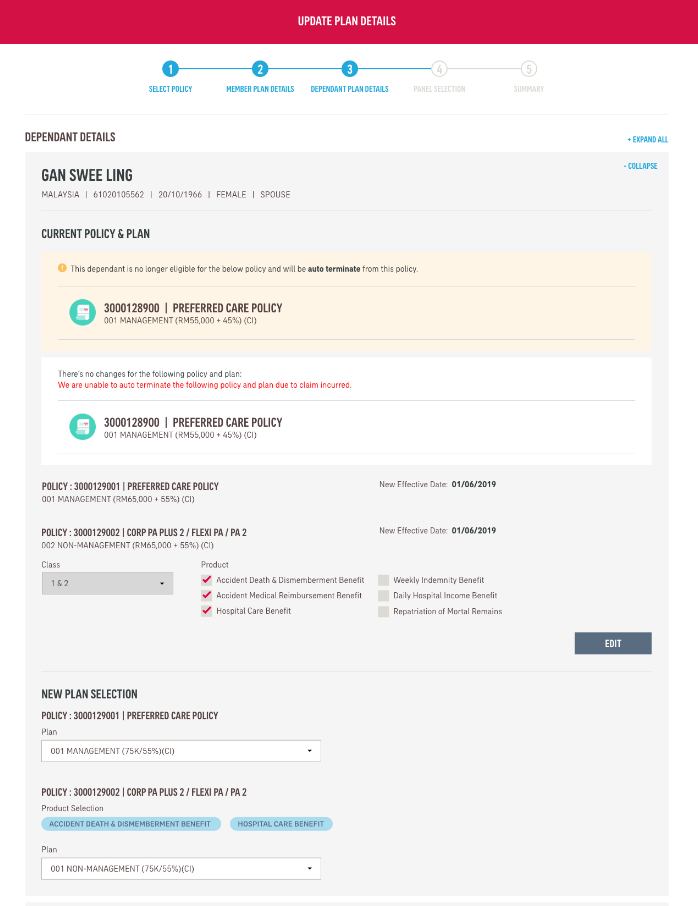
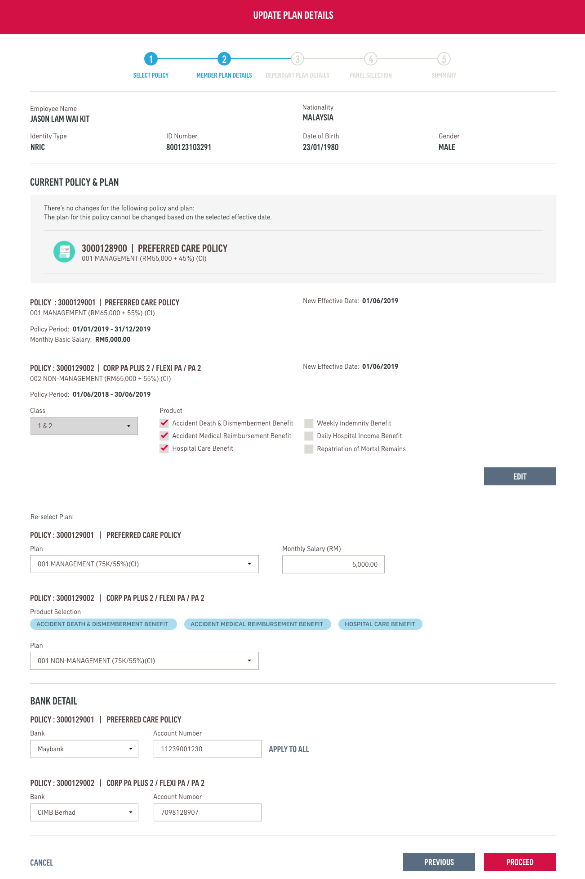
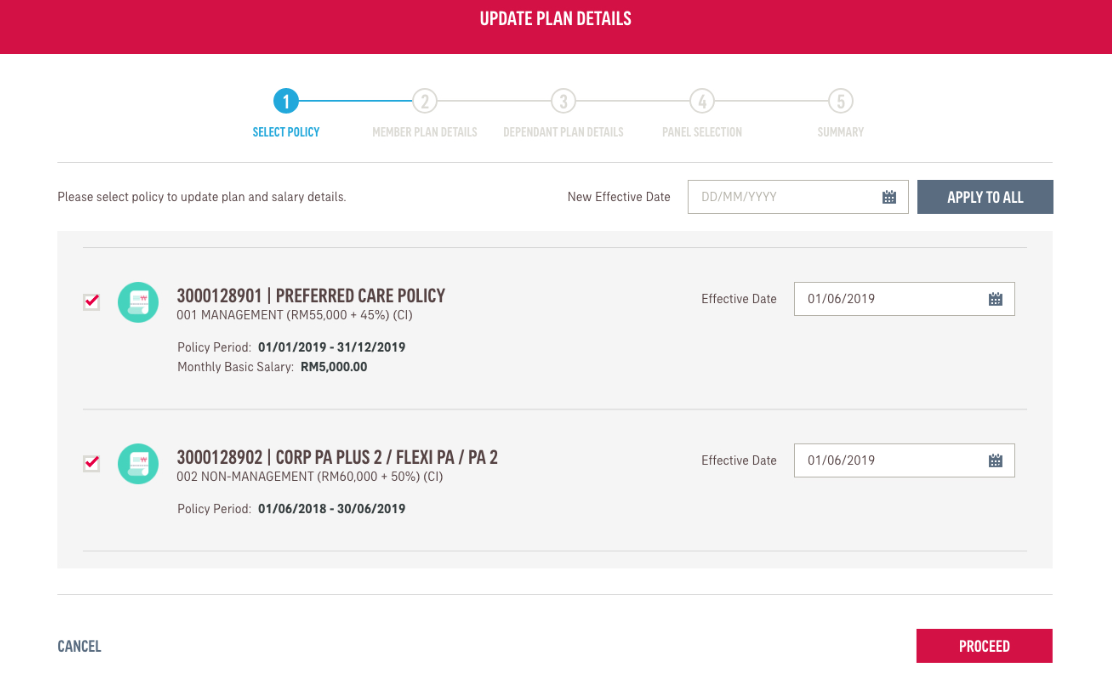
### Change Dependent Details Wireframes

The change dependent details wireframes are as below.



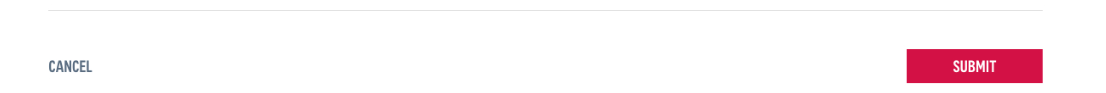
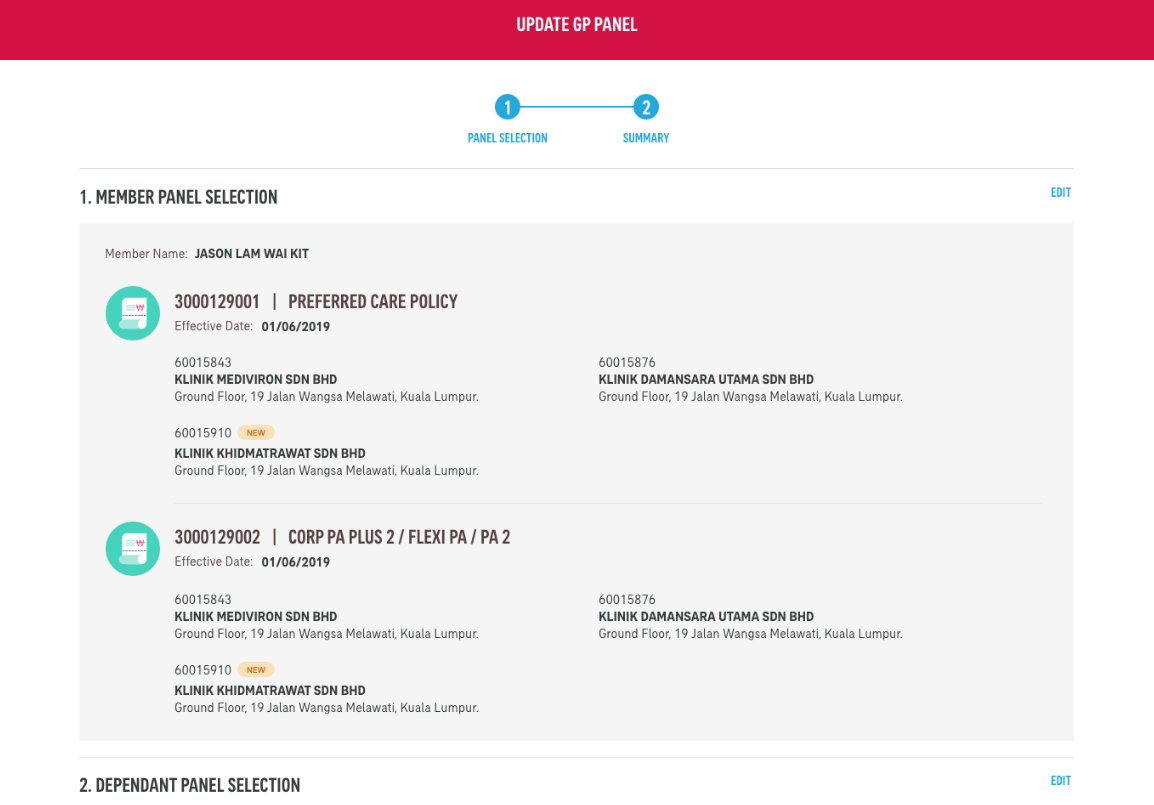
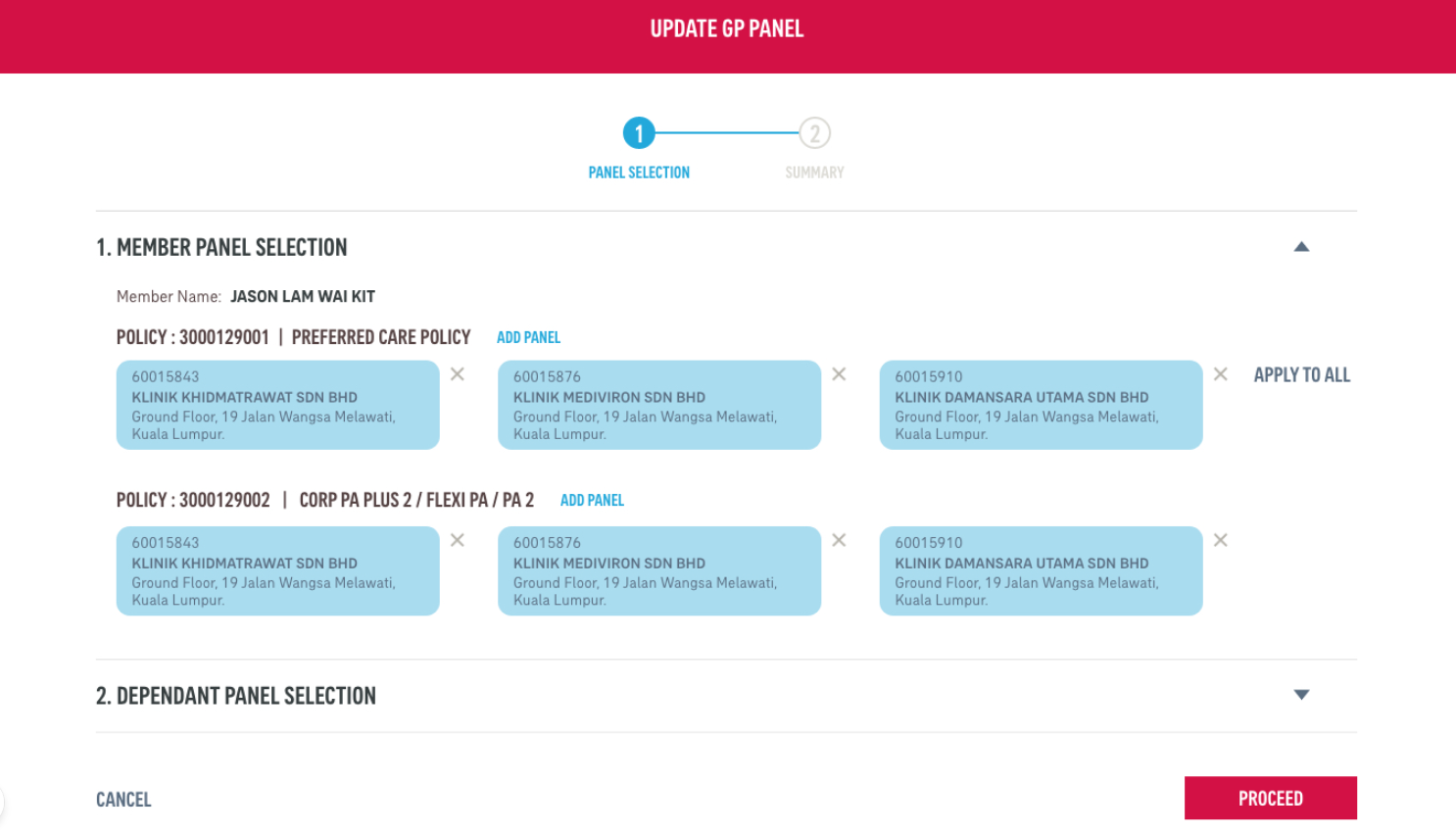
### Change Plan and Salary Wireframes

The change plan and salary wireframes are as below.



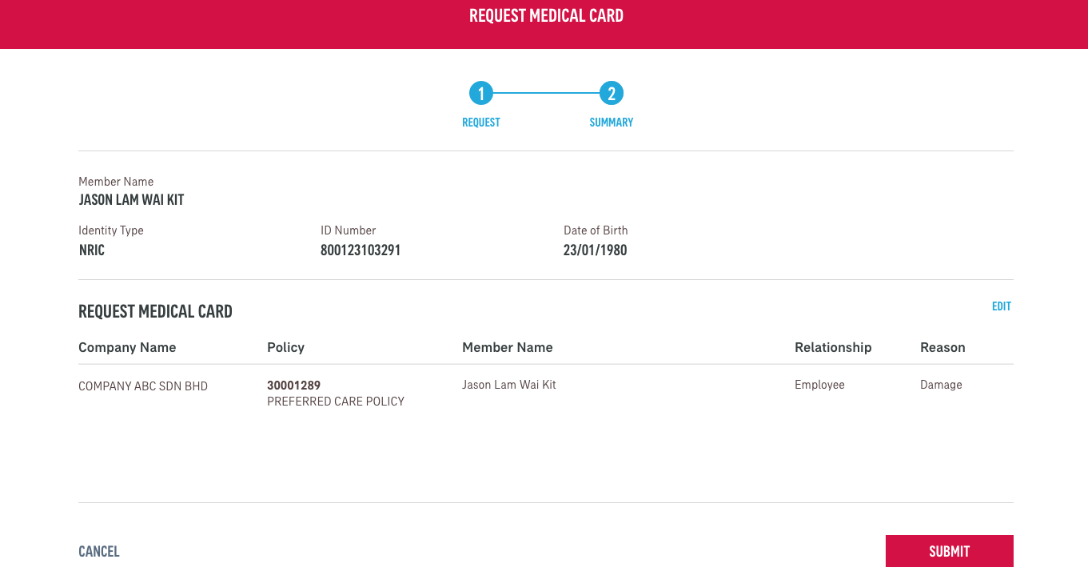
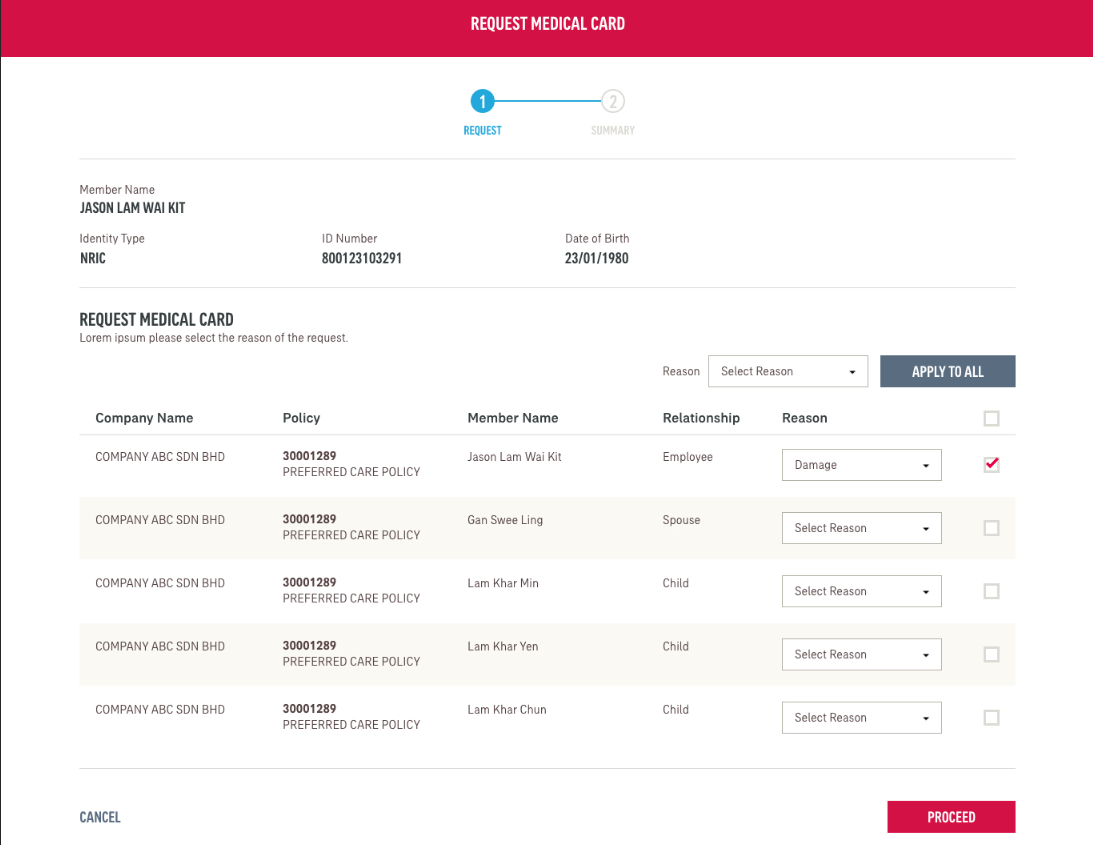
### Change Restricted Panel Wireframes

The change restricted panel wireframes are as below.



### Request Medical Card Wireframes

The request medical card wireframes are as below.



### Bulk Member Termination Wireframes

The bulk member termination wireframes are as below.

