User Manual
Child Sponsorship and Analytics for AMG Guatemala

To be submitted to the Department of Mathematics and Computer Science
Gordon College
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

by

Jacob Buettner
Joshua Richard
Dane Vandenberg

Revision date: December 13th, 2015

Document accepted on December 13th, 2015 by _____

Document accepted on _____ by _____

Use Case 1: Sponsor Child

Actors: Initiated by User

Notifies AMG Guatemala admin

Flow of Events:

1. User selects one or more children from the the website to sponsor
    1. Webpage queries MongoDB to see if the child is unsponsored, if they are the child document is locked and the user is forwarded to a checkout page
    2. If child is sponsored or locked, the user is alerted and asked to select another child to sponsor
2. On the checkout page the user can see the children they have selected to donate and a form to enter their data.
3. When the user submits the checkout form the form is sent to the node server, which sends the financial information to the bank and waits for a confirmation from the bank
    1. If confirmation is received from the bank, the MongoDB is updated and the user is sent to a sponsorship confirmation page
        a. Notification sent to AMG Guatemala admin that a child/children have been sponsored
    2. If the transaction is rejected by the bank or no confirmation is received from the bank, the user is alerted that we were unable to process their donation and to please try again later.

Entry Conditions: The user logged into their account

Exit Conditions: Confirmation is received from the bank, the mongo database is updated, the user has been notified that they are now sponsoring their selected child/children, and AMG Guatemalan admins have been notified that the child/children have been sponsored

Use Case 2: Cancel Sponsorship

Actors: Initiated by User

Notifies AMG Guatemala admin

Flow of Events:

1. User navigates to their account page on the AMG Guatemala website
2. The user selects the donation/sponsorship they would like to cancel
    a. User selects cancel
        i. User will be prompted to reenter some/all bank information so that we can contact the bank.
3. Request is sent to the node server
    a. Node server sends request to Bank to cancel selected payment
        i. Confirmation received from Bank, MongoDB is updated, user is notified, AMG Guatemala admin is notified
        ii. No confirmation received from Bank and user is notified we were unable to process their request and to please try again later or contact an AMG Guatemala admin for assistance

Entry Condition: User is logged into their account

Exit Condition: Sponsorship has been canceled and the user and AMG Guatemala admin has been notified of the cancellation.
Quality Requirements: If we cannot process the users cancellation, they are given an option to contact an AMG Guatemala admin for further assistance.

Use Case 3: Modify/Edit donation

Actors: Initiated by User
Notifies AMG Guatemala admin
Flow of Events:

1. User navigates to their account page on the AMG Guatemala website
2. The user selects the donation/sponsorship they would like to modify
    a. User selects modify
        i. User will be prompted to reenter some/all bank information so that we can contact the bank.
3. Request is sent to the node server
    a. Node server sends modification
    b. Request to Bank
        i. Confirmation received from Bank, MongoDB is updated, user is notified, AMG Guatemala admin is notified
        ii. No confirmation received from Bank and user is notified we were unable to process their request and to please try again later

Entry Condition: User is logged into their account
Exit Condition: User has been notified of the modification made and an AMG Guatemala admin has been notified of the modification.
Quality Requirement: User can immediately view their updated sponsorships

Use Case 4: Create User Account

Actors: Initiated by User
Flow of Events:

1. User clicks on create account, or a user tries to sponsor a child without logging in first
2. User is directed to an account creation form
3. User fills out the form and clicks submit
    a. MongoDB updated with user form information

Entry Conditions: User does not already have an account
Exit Conditions: User account is created
Quality Requirements: User is logged into their new account

Use Case 5: Delete User Account

Actors: Initiated by User
Flow of Events:

1. User navigates to their account page on the AMG Guatemala website
2. User clicks delete account button
    a. User is asked to confirm deletion
    b. User submits form including credit information

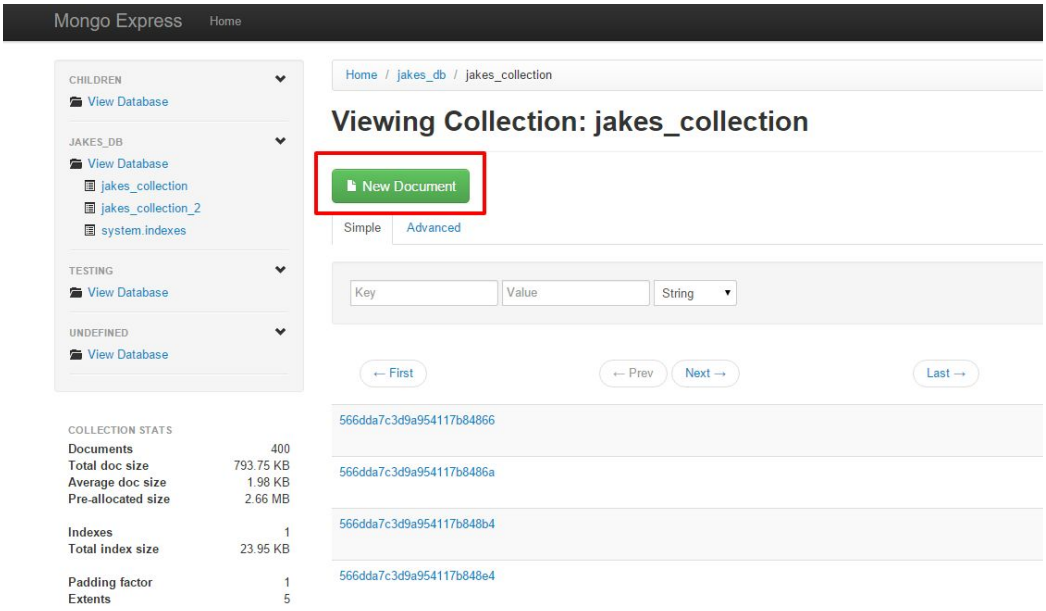
c. Request sent to Bank to cancel all payments from user
    i. Confirmation received from Bank, MongoDB is updated, AMG Guatemala admin is notified
    ii. No confirmation received from Bank and user is notified we were unable to process their request and to please try again later
3. User account is removed from MongoDB
    a. User is notified that their account has been deleted and all payments have been cancelled

Entry Conditions: User has an account and is logged in
Exit Conditions: User account has been deleted

Use Case 6: Add a child document to the database

Actors: Initiated by AMG Guatemala Admin
Flow of Events:

1. Admin can either add a document to the database using the Mongo shell or through Mongo-Express
    a. If using command line, admin will insert a document using insert()
2. If using Mongo-Express the admin will navigate to the web interface
    a. click new document



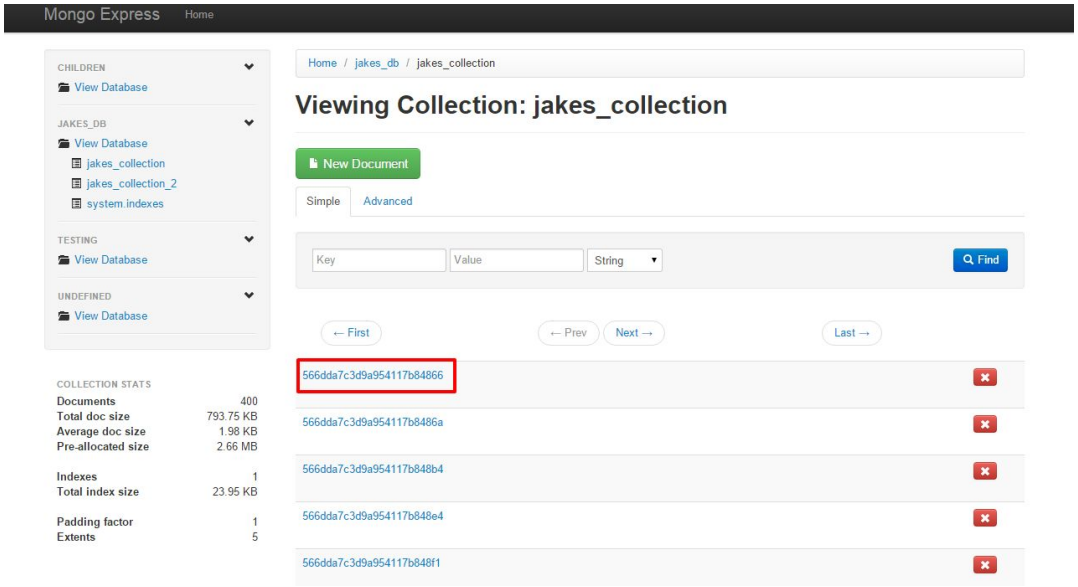
    b. Input information in JSON

Entry Conditions: There is an existing collection to insert the document into
Exit Conditions: Document has been added to the collection

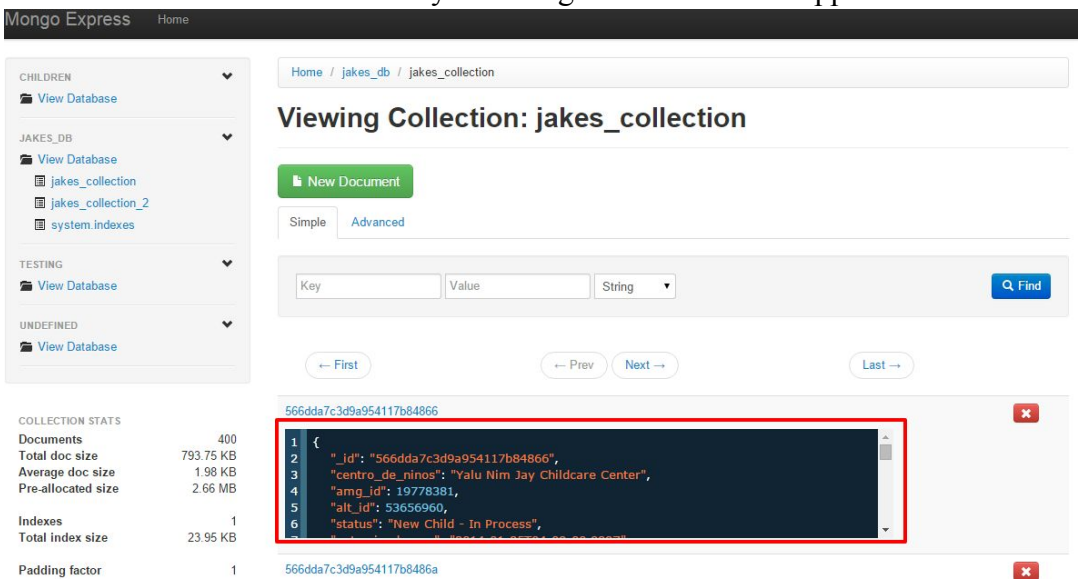
Use Case 7: Modify information on database

Actors: Initiated by AMG Guatemala Admin
Flow of Events:

1. Admin can either modify a document in the database using the Mongo shell or through Mongo-Express
    a. If using command line, admin will modify document using db.collection.update
2. If using Mongo-Express the admin will navigate to the web interface
    a. click on the document you would like to edit



b. Then make your changes in the box that appears

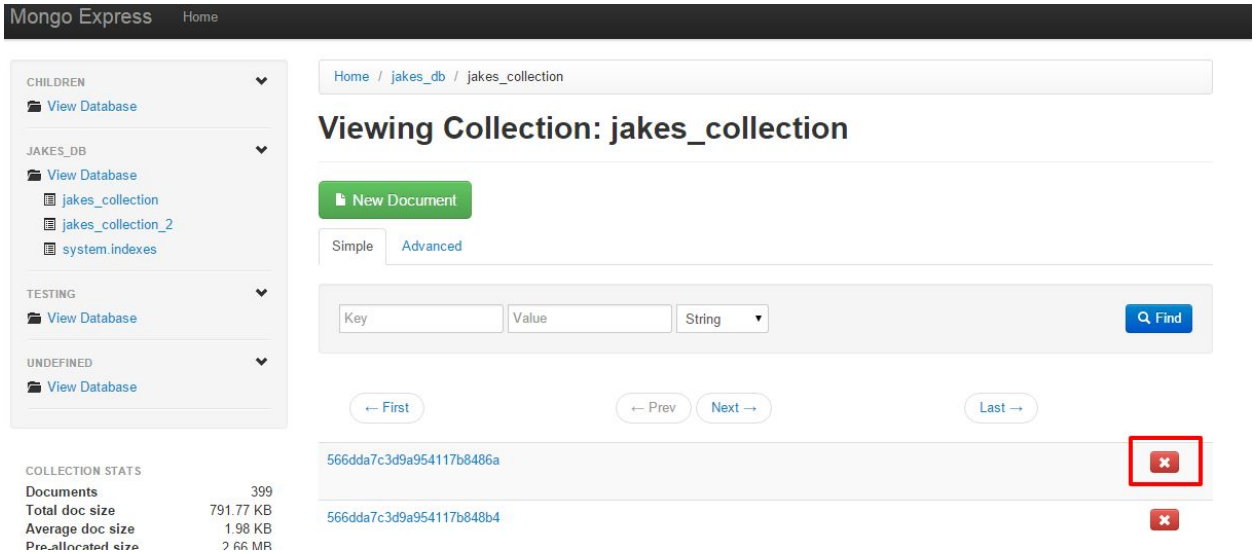


Entry Conditions: Admin is editing a document that exists
Exit Conditions: Document has been added modified and is still in the collection

Use Case 8: Delete information from database

Actors: Initiated by AMG Guatemala Admin
Flow of Events:

1. Admin can either delete a document in the database using the command line or through Mongo-Express
    a. If using command line, admin will modify document using the db.collection.remove command
2. If using Mongo-Express the admin will navigate to the web interface
    a. click the red "x" next to the document you would like to delete



Entry Conditions: Deleting a document that exists
Exit Conditions: Document no longer exists

Notes:

Child Document - A child document is the where all the information for a given child is stored. This is generally equivalent to a row in a table in a sql database. Information in a document is stored in JSON. Documents do not have a required structure as a row in a sql table does, but document of the same type should have similar structures. MongoDB will not enforce and structure so it is up to the developer to maintain document structure. An example child document may look something like this:


```
{
    "_id": ObjectId("56a50d00535c3be317bf51f5"),
    "amg_id": NumberInt(11196939),
    "alt_id": NumberInt(77571751),
    "image_id": ObjectId("5690af8462b84ed1d336c51a"),
    "status": "Waiting for Sponsor - No Prior Sponsor",
    "patrocinado_por": ISODate("2005-05-10T16:52:23.438Z"),
    "nombre": "Amanda",
    "segundo_nombre": "Amanda",
    "apellido": "Jimenez",
    "género": "masculino",
    "cumpleaños": ISODate("2002-06-02T03:02:47.608Z"),
    "años": NumberInt(13),
    "centro_de_ninos": "Filadelfia Childcare Center",
    "direccion_de_casa": "585 Gxvwziffhbkjvt Drive",
    "ciudad": "Antigua Guatemala",
    "provincia": "Escuintla",
    "código_postal": NumberInt(33182),
    "patrocinador_id": NumberInt(46329),
    "patrocinador_nombre": "Lucia",
    "última_actualización": ISODate("2015-03-07T02:15:44.705Z"),
    "abscent_padre": "true",
    "abscent_madre": "false",
    "religión_de_la_familia": "Buddhism",
    "iglesia_de_la_familia": "Purple Church",
    "estado_civil_de_los_padres": "Divorced",
    "comidas_al_día_en_el_hogar": NumberInt(0),
    "guardians": {
      "guardian_0": {
        "relacion": "abuelo",
        "nombre": "Amanda",
        "apellido": "Reyes",
        "ocupación": "vet",
        "ingreso_mensual_en_usd": NumberInt(91341),
        "cumpleaños": ISODate("1978-05-19T13:03:43.560Z")
      }
    },
    "hermanos": {
      "hermanos_0": {
        "nombre": "Tomas",
        "apellido": "Jimenez",
        "cumpleaños": ISODate("2000-02-23T11:43:50.641Z"),
        "grado_u_ocupación": NumberInt(10),
        "género": "masculino",
        "status": "New Child - In Process",
        "amg_id": NumberInt(22504175),
```

Collection - A collection is where documents are stored. A collection is generally equivalent to a table in a sql database.